



引文格式:郑晔,郭仁忠,贺彪,等.利用映射-归约的分布式区域对象可视查询方法[J].武汉大学学报(信息科学版),2023,48(9):1482-1489.DOI:10.13203/j.whugis20210133

Citation: ZHENG Ye, GUO Renzhong, HE Biao, et al. Distributed Visible Query Method for Regional Objects Using Map-Reduce[J]. Geomatics and Information Science of Wuhan University, 2023, 48(9): 1482-1489. DOI: 10.13203/j.whugis20210133

利用映射-归约的分布式区域对象可视查询方法

郑晔^{1,3} 郭仁忠^{1,2} 贺彪^{1,2} 马丁^{1,2} 李晓明^{1,2} 赵志刚^{1,2}

1 深圳大学建筑与城市规划学院智慧城市研究院,广东 深圳,518060

2 深圳市空间信息智能感知与服务重点实验室,广东 深圳,518060

3 自然资源部国土空间大数据工程技术创新中心,广东 深圳,518060

摘要:区域对象的可视查询是沿着区域视线方向剔除三维空间数据中一些表面被遮挡住而导致不可见的对象,从而提高大规模三维场景的可视渲染效率。针对传统区域可视查询视点空间划分粒度细、计算难度大的现状,提出了一种基于映射-归约的分布式可视查询方法。在映射函数中,按照三维对象的空间分布规律构建层级轴对齐包围盒,以轴对齐包围盒为视点空间划分区,将可视域范围内三维对象发送至规约函数中进行可视判断。在规约函数中,利用平面拆分后射线求交方法,通过构建二叉空间分割树计算每一视点空间划分区的潜在可视集,从而实现三维空间对象的分布式可视查询。将此方法用于深圳市20多万个三维空间对象的可视查询实验中,从数据量、划分粒度和并行度等角度验证了算法的可行性和有效性。

关键词:可视查询;映射-归约;空间分布式计算;三维地理信息系统;智慧城市

中图分类号:P208

文献标识码:A

收稿日期:2021-08-10

DOI:10.13203/j.whugis20210133

文章编号:1671-8860(2023)09-1482-08

Distributed Visible Query Method for Regional Objects Using Map-Reduce

ZHENG Ye^{1,3} GUO Renzhong^{1,2} HE Biao^{1,2} MA Ding^{1,2}

LI Xiaoming^{1,2} ZHAO Zhigang^{1,2}

1 Research Institute for Smart Cities, School of Architecture and Urban Planning, Shenzhen 518060, China

2 Key Laboratory of Spatial Information Smart Sensing and Services, Shenzhen 518060, China

3 National Engineering Laboratory for Big Data System Computing Technology, Shenzhen 518060, China

Abstract: Objectives: In the large-scale of virtual reality scene, it is difficult to add all graphics data into the video memory for rendering. Removing the occluded objects in advance by visible query technology can reduce the amount of data loaded on the display end to improve the rendering efficiency. Therefore, the research of visible query method for regional objects has important application value for real-time rendering of large-scale urban scene. **Methods:** We put forward a distributed visible query method based on Map-Reduce. In the map phase, we apply a hierarchical axis-aligned bounding box as viewpoint space partition. When the number of 3D objects in viewpoint space partition exceeds the threshold, the axis-aligned bounding box continues to be divided into sub-boxes. After the above process, the map tasks produce Geo-Tuples with the VSPID as key and visible query candidate set as value. In the reduce phase, a viewpoint is created for each leaf axis-aligned bounding box where the binary space partitioning trees are build and the visible set is calculated using real-time occlusion algorithm. **Results:** The experiment results with a building compound, containing more than 200 000 geometric solids, in Shenzhen, China show that: (1) There is no simple linear relationship between the running time of distributed visible query and the number of viewpoint space partitions. (2) Running time and parallelism are not simply inversely proportional. The computational

基金项目:国家自然科学基金(42201449,41971341);国家重点研发计划(2019YFB2103100);浙江省自然科学基金(LQ23D010005)。

第一作者:郑晔,博士,主要研究领域为时空大数据处理、三维数据分布式管理。zhengye@szu.edu.cn

通讯作者:赵志刚,博士,副教授。zhaozgrisc@szu.edu.cn

efficiency of each process first increases and then decreases with the increase of parallelism. About 48 parallelism, the process has the highest efficiency. (3) Whether the distributed approach is better than the traditional approach depends on the number of 3D objects. After the amount of 3D objects reaches about 40 000, the distributed algorithm begins to be better than the traditional algorithm. **Conclusions:** The computational experiments reveal the proposed algorithms outperform competitors in terms of the processing efficiency and feasibility, which can meet the requirement of visible query in large-scale scenarios.

Key words: visible query; map-reduce; distributed spatial query; three-dimensional geographic information system; smart city

虚拟现实是现实世界在数字环境中的映射,有助于增强人对现实世界理解与认知,在智慧城市的建设和发展中扮演着重要的角色,也是泛地图学研究的重要内容^[1-2]。在虚拟地理场景中查找可见地理实体一般采用三维空间数据可视查询技术^[3],即以视点为原点沿视锥两侧延展构建可视区域,去除区域内不可见对象并获取可见数据集。在大规模虚拟场景内,三维空间数据量庞大,将数据全部加入显存进行图形渲染难以实现。利用可视查询预先将被遮挡物体剔除,在渲染过程中实时调用计算后的可见物体,可以减少显端加载数据量以提高渲染效率。因此,开展三维空间数据可视查询研究对于实现大规模城市场景的实时渲染具有重要的应用价值。

目前,三维空间数据的可视查询算法可以分为视点对象可视查询和区域对象可视查询^[4]。视点对象可视查询在显存中构建空间索引后,通过计算机图形学中三维消隐算法实现。区域对象可视查询是将场景空间划分成若干离散视点空间划分区,以每一视点空间划分区的几何中心作为视点后分别对其进行视点对象可视查询。在城市虚拟环境中所涉及场景大多为区域对象可视查询,计算过程中需要耗费大量的时间和空间资源(文献[5]表明存在 n 个空间对象的三维场景中,区域对象可视查询的时间复杂度为 $O(n^9(\log n))$,空间复杂度为 $O(n^9)$)。另一方面,随着测绘技术提高、数据采集能力增强,城市三维空间数据精细度和体量迅猛增加,单节点的计算资源难以承受可视查询中的存储和计算压力。

面对海量空间数据计算难题中,将空间数据划分成多个子域后通过技术手段实现多个空间子域数据的并行计算是一个有效的解决途径^[6-7]。分布式空间计算是将计算任务分发至不同的节点实现空间算法的并行化。当前,空间数据分布式计算(也称高性能计算)^[8]主要借助于开源大数据平台,通过合理的空间数据划分实现计算任务

的负载均衡,广泛应用于瓦片构建^[9-10]、空间分析^[11-12]、遥感图像处理^[13-17]等等各类地学计算中。然而,上述空间计算一般都是针对于二维矢量或栅格数据进行分布式计算,对于三维空间数据,尤其三维数据的可视并行查询领域目前展开研究工作相对较少。具有代表性的是 Hladky 等^[18]提出的采用渲染管道和计算管道相结合的并行算法。然而,上述算法主要基于单节点 CPU (central processing unit) 或 GPU (graphics processing unit) 上实现多线程。受限于计算资源,在数据总量较大时,空间数据结构在内存中维护难度大,因此具有一定局限性。

本文借鉴映射-归约计算模式(本方法不限于 Hadoop 的 MapReduce 框架实现,采用中文术语映射-归约来表达)思想将各视点空间划分区中的可视查询算法分布至不同节点中并行计算,从而实现大场景下分布式区域对象可视查询以提高计算效率。将此方法应用于深圳市香蜜湖片区的精细模型实验,结果表明,此方法能够弥补上述研究的不足,具有较高的实际应用价值。

1 相关术语

定义 1: 点的可视性。对于视点 P 、点 Q 和障碍物集合 O ,不存在 $o \in O$ 且 o 在 PQ 的连线上,即 $[P, Q] \cap o \in \emptyset$,则称点 Q 对视点 P 是可见的^[3]。

定义 2: 三维空间对象的可视性。对于视点 P 、三维对象 V 和障碍物集合 O ,存在 V 上的点 q 使得 q 对视点 P 是可见的,对于 $\forall o \in O$,使得 $[P, q] \cap o \in \emptyset$,则称三维对象 V 对视点 P 是可见的^[19-20]。

定义 3: 潜在可视集。由于准确计算可见集会耗费大量的计算,通常通过一种近似的、计算量更小的检索算法实现三维图形粗略的剔除,未被剔除的三维空间数据集被称作为潜在可视集^[21]。

定义 4: 视点空间划分区。视点空间划分区

(viewpoint space partition, VSP)表示在分区内,从任意视点观察三维对象,所有的三维对象的拓扑关系保持不变^[22]。由于始终维持相同的拓扑关系,因此该区域内所有视点的可视集保持不变。在现实中,每个视点划分都必须执行可视检索运算,计算结果所得潜在可视集形成图像帧。

定义5:轴对齐包围盒和有向包围盒。如图1(a)所示,三维对象的轴对齐包围盒(axis-aligned bounding box, AABB)表示能容纳该三维对象且平行于坐标轴的最小六面体。由于AABB与坐标轴平行,在近似代替三维对象时留有较多空白。如图1(b)所示,有向包围盒表示在三维空间中,能够容纳三维对象的最小包围盒^[23]。与轴对齐包围盒不同,有向包围盒通常不完全平行于世界坐标轴。因此,有向包围盒能更好地反映三维对象的空间占用情况,广泛应用于体碰撞和相交计算中。一般用中心点坐标和3个方向的半长轴表达有向包围盒。

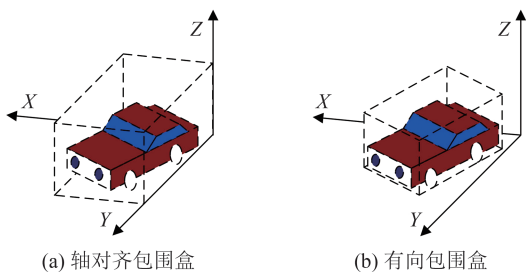


图1 两种包围盒的示意图

Fig. 1 Illustration of Two Kinds of Bounding Boxes

2 方法

区域对象的可视查询方法是三维空间场景划分成多个视点空间划分,每一视点空间划分应用遮挡剔除算法实现整个区域的可视查询。当三维场景中空间对象结构、分布较为复杂时,所划分的视点空间划分粒度细、数量多,导致算法时间和空间复杂度都很大。各个VSP之间的可视查询是相互独立且无依赖的计算过程,所以此算法本质上是可并行的。基于上述推论,本文提出一种基于映射-归约的三维空间对象的分布式可视查询方法。算法总流程如图2所示。

其主要包括以下步骤:

1)统一的三维模型构建。对多源三维数据(包括精细模型、倾斜摄影和点云数据)进行抽取并构建统一化的键值对元组(后文将此键值对模型称为GeoTuple)。GeoTuple包括几何信息、

纹理信息和材质信息。纹理信息和材质信息作为以二进制格式存储于GeoTuple中;几何信息包括三维对象唯一标识信息,所对应的三角面、顶点等几何信息和有向包围盒。键值对元组是云环境下NoSQL数据库中最基本的存储组织单元,将上述几何信息与外观信息(包括纹理和材质)组合形成三维地理元组(GeoTuple)。在分布式存储系统中,GeoTuple在主节点的协调下,通过各个计算节点执行节点间迁移过程,并实现节点间存储负载均衡和副本均衡。

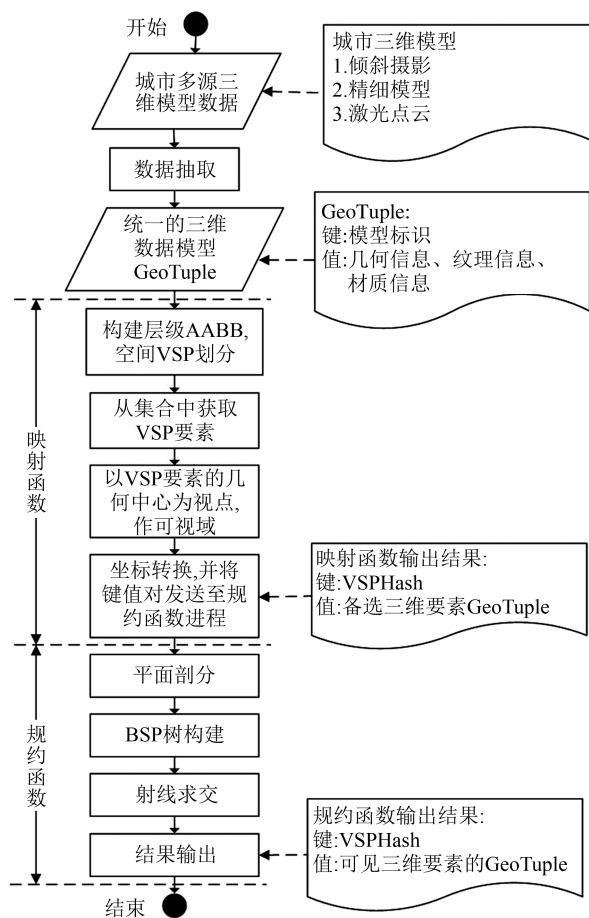


图2 算法总流程图

Fig. 2 Flowchart of Algorithm

2)基于映射-归约的可视查询。基于映射-归约计算模式的三维对象的可视查询具体步骤如下:在映射函数中,构建层级轴对齐包围盒作为VSP将整个三维空间划分成各个子域;以VSP几何中心向外扩大一定距离作可视域,所有与视域范围具有相交关系的三维对象称为此VSP的可视候选集。映射函数将每一VSP可视候选集的GeoTuple键值对输出至规约函数。规约函数得到每一视点的可视集后,首先按照文献[24]思想,将三维空间平面划分成若干个子平面,从而

得到三维对象在每个子平面上的二维投影;然后通过射线求交法^[25]计算每一投影面上二维数据的可见集;最后对上述投影面所得结果汇总得到最终的潜在可视集。

2.1 层级轴对齐包围盒构建

如图 3 所示,本文采用层级轴对齐包围盒实现 VSP 的划分,层级 AABB 是一种具有层次结构的 AABB。层级 AABB 构建过程中,当父级 AABB 存储数据超过一定数量时,将会分裂成若干子级 AABB。

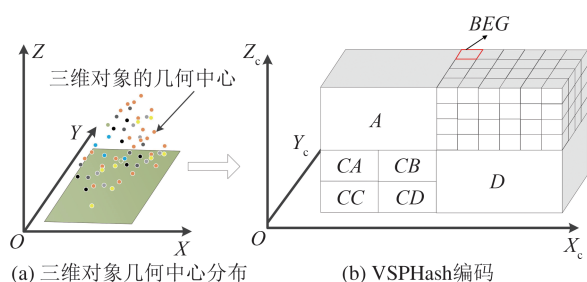


图 3 层级轴对齐包围盒示意图

Fig. 3 Illustration of Hierarchical Axis-Aligned Bounding Box

层级 AABB 构建方法流程如下:

1) 通过所有三维对象的边界范围,定义层级 AABB 的根节点。

2) 按照均等化的划分方式,将父级 AABB 划分成多个大小为 $L \times W \times H$ 的子级 AABB。

假设父级 AABB 空间范围坐标为 $[X_{\min}, X_{\max}, Y_{\min}, Y_{\max}, Z_{\min}, Z_{\max}]$, 则每一子级 AABB 的空间范围坐标为:

$$\begin{aligned} \text{AABB}(r, c, h) = & \left[\frac{X_{\max} - X_{\min}}{L} \times \right. \\ & r + X_{\min}, \frac{X_{\max} - X_{\min}}{L} \times (r + 1) + X_{\min}, \\ & \frac{Y_{\max} - Y_{\min}}{W} \times c + Y_{\min}, \\ & \frac{Y_{\max} - Y_{\min}}{W} \times (c + 1) + Y_{\min}, \\ & \frac{Z_{\max} - Z_{\min}}{H} \times h + Z_{\min}, \\ & \left. \frac{Z_{\max} - Z_{\min}}{H} \times (h + 1) + Z_{\min} \right] \end{aligned} \quad (1)$$

式中, $r \in [0, L]$; $c \in [0, W]$; $h \in [0, H]$ 。

3) 不断迭代步骤 2), 直至叶子节点。对于任意三维对象, 可用层级 AABB 的叶子节点表达, 假设三维对象中心点为 $P(x, y, z)$, 叶子 AABB 的计算方式如下:

$$r = \left\lfloor \frac{x - X_{\min}}{L} \right\rfloor, c = \left\lfloor \frac{y - Y_{\min}}{W} \right\rfloor, h = \left\lfloor \frac{z - Z_{\min}}{H} \right\rfloor \quad (2)$$

式中, $\lfloor \cdot \rfloor$ 表示向下取整。

4) 对步骤 3) 中的叶子 AABB, 利用空间填充曲线进行编码并赋予唯一 VSPHash, 即用字符串表达 AABB 在三维空间中的具体位置。VSPHash 的字符串的层级关系由字符串表达, 任意子级 AABB 的子节点编码均是以父级节点 VSPHash 值开头, 例如, 图 4 中编号为 B 的父级 AABB 分裂时, 所有的子级 AABB 均以 B 开头。

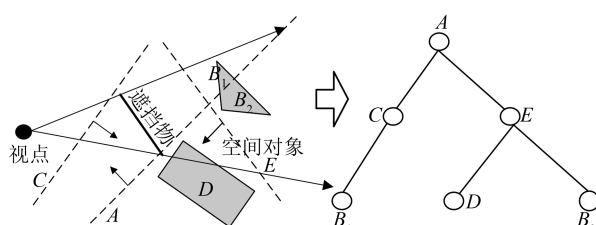


图 4 二叉空间分割树的构建

Fig. 4 Illustration of Construction of Binary Space Partitioning Tree

5) 判断 AABB 中的几何中心点个数是否超过最大阈值, 如果超过则继续递归执行步骤 2)、3)、4); 否则, 函数结束。

通过层级 AABB 可以将整个空间划分成若干个子域, 每一子域视作为三维空间的 VSP。

该做法具有以下优点: (1) 空间范围内, 三维对象较多时其空间拓扑关系就更复杂, 在此区间内能维持相同拓扑关系的 VSP 区间则更小, 因此需要空间划分的粒度就更小; 反之, 则相反, 这和层级立方体的组织原则相符。 (2) 层级 AABB 采用 Hash 编码方式, 可以通过字符串识别三维对象的空间位置。一方面根据 VSPHash 长度判断空间对象的密集程度; 另一方面后续步骤中可视域的三维空间查询中可以将相交运算转换成字符串匹配问题, 减少计算复杂度。

上述 VSP 划分方式可以将三维空间划分为多个子域 VSP, 每一个 VSP 为独立的 AABB。

映射函数步骤如下: (1) 遍历每个子域 VSP, 获取子域 VSP 的 Hash 值和几何中心; (2) 以 VSP 的几何中心为球心, 以一定的阈值半径作可视域球 (无论是否被遮挡, 该 VSP 视点对所有可视域球范围外的三维对象均不可见); (3) 输出所有与可视域球具有相交关系的三维对象, 并以 GeoTuple 形式传入规约函数中。

2.2 VSP内的可视查询

映射函数将几何体按照 VSP 的 Hash 值为键和 GeoTuple 为值,作为输入传入规约算法。规约算法可以把所有同一 VSP 阈值半径内的三维对象 GeoTuple 规约至同一进程,实现 VSP 内的可视查询。

VSP 内的可视查询方法流程如下:(1)按照文献[25]分别计算 VSP 内所有要素在切平面上的投影映射,从而获得多个切平面上的二维平面图形的集合;(2)如图 4 所示,分别对每一组切平面图形集构建二叉空间分割树,生成一组 BSP 树,BSP 树的构建方式参照文献[26];(3)在每一平面投影上,基于每个视点立方体的中心向可视域范围内 360° 的每一度方向做射线,并利用步骤(2)中的 BSP 树,计算与该射线具有相交关系的二维图形,返回距离视点最近的二维图形所对应的三维几何图形;(4)分别在各个切平面计算步骤(3)中并将结果求并集后去重,得到的即为该可视域立方体的可见集;(5)以视点立方体的 VSPHash 为键,以步骤(4)中可见集的 GeoTuple 为值,存入 NoSQL 数据库中。

2.3 算法的分布式实现

本文算法通过映射-归约模式实现分布式计算。映射-归约计算模式源自于函数式编程语言,在映射函数中按键将键值对元组并发地发送归约函数中,保证所有具有相同键的键值对元组进入同一规约函数进程中。

Map-Reduce 计算引擎是最早应用映射-归约模式实现海量存储于硬盘^[27-30],造成大量 I/O 操作降低计算效率。Spark 在针对 Hadoop 框架中的不足和缺陷,将 Transform 算子(包括映射和规约算子)的计算结果存放在内存中,直到执行 Action 算子时才实现数据向外输出,从而减少迭代过程中的数据落地,能够实现数据高效共享。视查询的映射-归约计算模式如图 5 所示。

本文的分布式算法实现示例如下:(1)映射函数将存储于 HDFS 中的三维模型 GeoTuple 并行读入本文方法中,并将本地中间结果以键值对元组 $\langle \text{VSPHash}, \text{GeoTuple} \rangle$ 输出;(2)当映射函数结束后,管理进程向工作节点发送洗牌命令,使得每个 VSPHash 相同,即同一 VSP 分区的键值对进入同一规约函数;(3)规约函数在接收到同一 VSP 分区三维对象的 GeoTuple 后,执行可视查询,并将最终结果存入 NoSQL 数据库中。

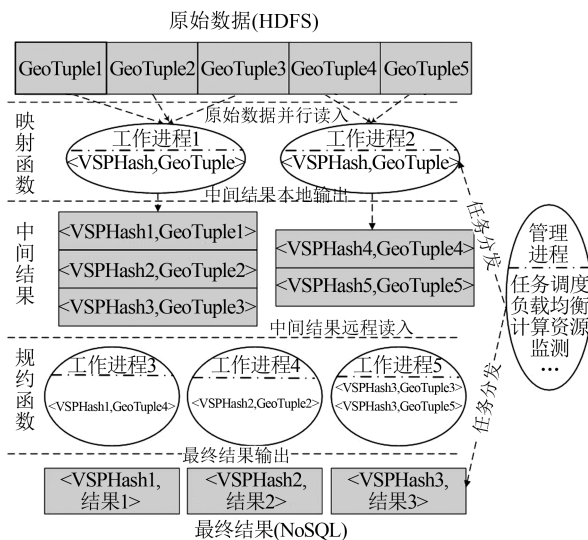


图5 视查询的映射-归约计算模式

Fig. 5 Map-Reduce Computing Paradigm for Visible Query

3 实验与讨论

3.1 实验环境和实验数据

本文实验的硬件环境包括5个计算节点,每个计算节点含有24核Intel Xeon(R) 2.3 GHz CPU以及64 GB内存。实验分配每个计算任务为含有2 GB内存的单核任务,因此,本实验最多可拥有80个进程并发量。实验部署于64位Linux操作系统CentOS 7中,其中分布式计算引擎采用Hadoop 3.1.2和Spark 2.4.3,存储引擎采用HBase 2.2.0作为NoSQL存储库,Redis 5.0作为分布式缓存引擎的双层架构,实验算法由Java 1.8实现。

如图6所示,实验区域选择为广东省深圳市福田区香蜜湖片区,其地理范围约为 $22.51^\circ\text{N} \sim 22.58^\circ\text{N}$, $113.98^\circ\text{E} \sim 114.10^\circ\text{E}$,此区域为深圳市中心城区,具有较为密集的城市建筑群。实验数据源格式为3DMAX模型(数据来源于深圳市可视化城市空间数字平台原型系统项目),全空间范围为长8 542 m \times 宽4 877 m \times 高440 m的长方体,共包含209 850个三维空间对象。

3.2 实验结果分析

为了验证本方法的运算效率,进行3组实验以探究不同因素下可视查询的运行时间。

在第1组实验中,本文更改两个实验变量(实验并行度为80):(1)层级AABB中叶子节点立方体的划分粒度;(2)可视域球的半径。探究改变上述实验变量条件下,可视查询并行算法的计算效率。其中,第1个实验变量采用VSP的平均边

长代替平均三维对象的个数,以便更好地反映 VSP 的空间划分程度。

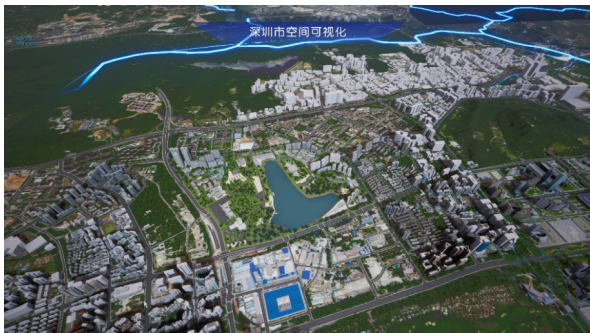


图 6 实验区范围概览图

Fig. 6 Overview of the Study Area

图 7 为 VSP 的平均边长和可视域球半径对算法效率的影响结果图。

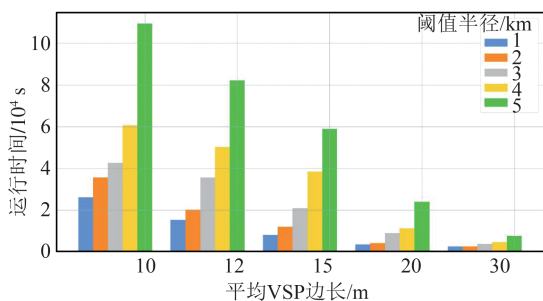


图 7 VSP 的平均边长和可视域球半径对算法效率的影响

Fig. 7 Influence of Average VSP Size and Threshold Radius on Runtime

理论上,可视查询计算时间复杂度为全空间范围的体积和叶子立方体的体积的比值,叶子立方体的平均边长每减少至原有的 $1/n$,可视计算时间复杂度将增加至原来的 n^3 ,当平均 VSP 边长设置 10 m 时生成大约 1 833 万帧潜在可视集。然而研究发现,随着 VSP 边长的减少,运行时间并不是简单地呈三次方关系。其原因是数据非均衡,尽管构建层级 AABB 的过程中考虑 VSP 内三维对象平均个数尽量均衡,然而当 VSP 向外扩展成可视域球时,无法保证可视域球内的三维对象是均衡的。如果每次计算可视域球的三维对象数量严重倾斜,将造成任务划分的不均衡,因此每一规约算法执行时间长短不一。

第 2 组实验将阈值半径设为 5 km,平均 VSP 边长设为 30 m,探究运行时间和并行度之间关系,结果如图 8 所示。

由图 8 可以发现,随着并行度的增加,更多计算资源参与可视查询,因此计算时间总体呈减少趋势。运行时间和并行度并不是呈简单反比关

系,单位并行度的计算效率随着并行度先增加后减少,大约在 48 个并行度左右,单位进程的计算效率最高。其原因如下:(1)在规约函数中,并行度增加减少了每个进程的数据量。尽管 BSP 构建和查询平均时间复杂度为 $O(N)$ (最优时间复杂度为 $O(\log N)$)^[26],但是数据量较少会导致深度增加,从而降低递归层数和栈帧,提高计算效率;(2)并行度的增加也会导致节点间的数据传输概率变高,从而降低计算效率。综上,并行度和运行时间的关系最终的结果取决于上述两个原因中起主导性的因素。

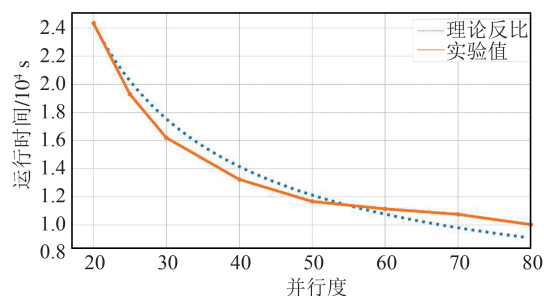


图 8 并行度和运行时间的关系

Fig. 8 Relationship Between Parallelism and Runtime

在第 3 组实验中,分别在原始数据中按空间分布分别抽取 2 万、5 万、10 万、15 万和 20 万个三维空间对象,将阈值半径设为 5 km,平均 VSP 边长设为 30 m,并行度设为 48,探究不同数据量下分布式算法和传统算法^[16]运行时间对比关系(为了提高实验的可靠性,分布式算法运行效率已换算成平均单进程的计算效率)。

从图 9 中可以得出,在数据量为 2 万三维空间对象时,传统算法计算效率更占优势。随着数据量增加,分布式算法的计算效率优势越来越得到体现,在数据量达到大约 4 万个三维空间对象时,分布式算法开始领先于传统算法。这是由于分布式计算中,任务的划分、数据分发、调度算法的实施都需要占用计算资源,而传统算法的时间复杂度始终为 $O(N^{1.5})$ 。在数据量不大时,传统单进程算法已经足够解决计算问题,应用分布式计算反而由于上述原因降低计算效率。当数据量到达一定程度后,分布式计算算法才能体现优势。

4 结 语

本文提出一种基于映射-归约的三维空间对象的分布式可视查询方法,将不同 VSP 间的可视查询任务派发至不同的计算机节点进程中,从而

提高计算效率。

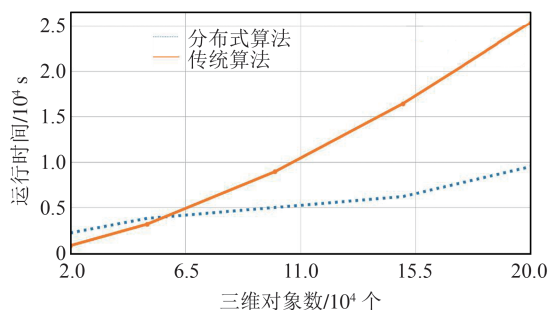


图9 同数据量下分布式算法和传统算法运行时间

Fig. 9 Runtime of Our Approach and Traditional Approach Under Different Data Volumes

实验结果表明,此方法具有非常好的时间和空间效率及可扩展性,能满足大规模场景下的可视计算需求。在规约函数中,本文采取平面切分后投影的方法计算VSP内部遮挡剔除,计算结果较为粗糙。在下一步研究工作中,将引入更精确的遮挡剔除算法,进一步提高检索的精度。

参 考 文 献

- [1] Guo Renzhong, Chen Yebin, Ying Shen, et al. Geographic Visualization of Pan-Map with the Context of Ternary Spaces[J]. *Geomatics and Information Science of Wuhan University*, 2018, 43(11): 1603-1610. (郭仁忠, 陈业滨, 应申, 等. 三元空间下的泛地图可视化维度[J]. 武汉大学学报(信息科学版), 2018, 43(11): 1603-1610.)
- [2] Guo Renzhong, Lin Haojia, He Biao, et al. GIS Framework for Smart Cities[J]. *Geomatics and Information Science of Wuhan University*, 2020, 45(12): 1829-1835. (郭仁忠, 林浩嘉, 贺彪, 等. 面向智慧城市的GIS框架[J]. 武汉大学学报(信息科学版), 2020, 45(12): 1829-1835.)
- [3] Kong Dehan, Liu Yongshan. Visible Query Based on Road Network in Three-Dimension Scene [J]. *Chinese Journal of Computers*, 2016, 39(10): 2045-2060. (孔德瀚, 刘永山. 基于路网的三维虚拟现实场景间接可视查询框架[J]. 计算机学报, 2016, 39(10): 2045-2060.)
- [4] Pantazopoulos I, Tzafestas S. Occlusion Culling Algorithms: A Comprehensive Survey[J]. *Journal of Intelligent and Robotic Systems*, 2002, 35(2): 123-156.
- [5] Clarisse P. Smart Cities in Japan: An Assessment on the Potential for EU-Japan Cooperation and Business Development [M]. Tokyo, Japan: EU-Japan Centre for Industrial Cooperation, 2014.
- [6] Zhou C, Chen Z J, Li M C. A Parallel Method to Accelerate Spatial Operations Involving Polygon Intersections[J]. *International Journal of Geographical Information Science*, 2018, 32(12): 2402-2426.
- [7] Fang Lei, Yao Shenjun, Bao Hangcheng, et al. An Algorithm for Optimizing Routing of Remote Sensing Image Parallel Processing Based on Data Partitioning[J]. *Acta Geodaetica et Cartographica Sinica*, 2019, 48(5): 572-582. (方雷, 姚申君, 包航成, 等. 遥感影像并行处理的数据划分及其路径优化算法[J]. 测绘学报, 2019, 48(5): 572-582.)
- [8] Zuo Yao, Wang Shaohua, Zhong Ershun, et al. Research Progress and Review of High-Performance GIS[J]. *Journal of Geo-Information Science*, 2017, 19(4): 437-446. (左尧, 王少华, 钟耳顺, 等. 高性能GIS研究进展及评述[J]. 地球信息科学学报, 2017, 19(4): 437-446.)
- [9] Yu Lijun, Zhang Feng, Liu Renyi, et al. A Spatial Indexing Method for Efficient Generation of Vector Tiles[J]. *Geomatics and Information Science of Wuhan University*, 2020, 45(10): 1633-1641. (俞丽君, 张丰, 刘仁义, 等. 一种面向矢量瓦片高效构建的空间索引方法[J]. 武汉大学学报(信息科学版), 2020, 45(10): 1633-1641.)
- [10] Nie Pei, Chen Guangsheng, Jing Weipeng. Parallel Construction and Distributed Storage for Vector Tile [J]. *Journal of Geo-Information Science*, 2020, 22(7): 1487-1496. (聂沛, 陈广胜, 景维鹏. 矢量瓦片并行构建与分布式存储模型研究[J]. 地球信息科学学报, 2020, 22(7): 1487-1496.)
- [11] He Dengping, He Zonghao, Li Peiqiang. A Parallel-high Utility Itemset Mining Algorithm Based on Spark[J]. *Computer Engineering & Science*, 2019, 41(10): 1723-1730. (何登平, 何宗浩, 李培强. 基于Spark的并行化高效用项集挖掘算法[J]. 计算机工程与科学, 2019, 41(10): 1723-1730.)
- [12] Zhang F, Zheng Y, Xu D, et al. Real-Time Spatial Queries for Moving Objects Using Storm Topology [J]. *ISPRS International Journal of Geo-Information*, 2016, 5(10): 178.
- [13] Wang Zhihua, Yang Xiaomei, Zhou Chenghu. Geographic Knowledge Graph for Remote Sensing Big Data[J]. *Journal of Geo-Information Science*, 2021, 23(1): 16-28. (王志华, 杨晓梅, 周成虎. 面向遥感大数据的地学知识图谱构想[J]. 地球信息科学学报, 2021, 23(1): 16-28.)
- [14] Qing Jianfei, Yang Yaping, Li Deping. Research on Method of Massive Remote Sensing Image Retrieval Based on Distribute ElasticSearch [J]. *Geomatics & Spatial Information Technology*, 2019, 42(6): 64-66. (卿建飞, 杨雅萍, 李德平, 等. 基于分布式ElasticSearch的大规模遥感影像检索方法研究[J]. 测绘与空间信息技术, 2019, 42(6): 64-66.)

- ticSearch的海量遥感影像检索方法研究[J]. 测绘与空间地理信息, 2019, 42(6): 64-66.)
- [15] Wu Huayi, Cheng Hongquan, Zheng Jie, et al. RS-ODMS: An Online Distributed Management and Service Framework for Remote Sensing Data[J]. *Geomatics and Information Science of Wuhan University*, 2020, 45(12): 1836-1846. (吴华意, 成洪权, 郑杰, 等. RS-ODMS: 一种分布式遥感数据在线管理与服务框架[J]. 武汉大学学报(信息科学版), 2020, 45(12): 1836-1846.)
- [16] Li J Y, Meng L K, Wang F Z, et al. A Map-Reduce-Enabled SOLAP Cube for Large-Scale Remotely Sensed Data Aggregation[J]. *Computers & Geosciences*, 2014, 70: 110-119.
- [17] Li Jiyuan, Gan Bin, Meng Linghui, et al. Rapid Imagery Tile Generation for Remotely Sensed Time-Series Data in the Cloud Environment[J]. *Geomatics and Information Science of Wuhan University*, 2015, 40(2): 243-248. (李继园, 甘斌, 孟令奎, 等. 云环境下时序遥感影像的快速缓存切片方法[J]. 武汉大学学报(信息科学版), 2015, 40(2): 243-248.)
- [18] Hladky J, Seidel H P, Steinberger M. The Camera Offset Space: Real-Time Potentially Visible Set Computations for Streaming Rendering[J]. *ACM Transactions on Graphics*, 2019, 38(6), DOI:org/10.1145/3355089.3356530.
- [19] Nutanong S, Tanin E, Zhang R. Visible Nearest Neighbor Queries [C]//Advances in Databases: Concepts, Systems and Applications, Berlin, Germany, 2007.
- [20] Sultana N, Hashem T, Kulik L. Group Nearest Neighbor Queries in the Presence of Obstacles[C]//Association for Computing Machinery, Dallas, Texas, 2014.
- [21] Luebke D, Georges C. Portals and Mirrors: Simple, Fast Evaluation of Potentially Visible Sets [C]//Symposium on Interactive 3D Graphics, New York, USA, 1995.
- [22] Chakravarty I, Freeman H. Characteristic Views as a Basis for Three-Dimensional Object Recognition [J]. *Proceedings of SPIE: The International Society for Optical Engineering*, 1982, 0336: 37-45.
- [23] O'Rourke J. Finding Minimal Enclosing Boxes[J]. *International Journal of Computer & Information Sciences*, 1985, 14(3): 183-199.
- [24] Gu Weijie, Liu Yongshan. Research on Directional Relation Combinational Reasoning of Double Projections Matrix Model [J]. *Journal of Geomatics Science and Technology*, 2014, 31(5): 538-542. (顾卫杰, 刘永山. 双投影矩阵模型的方向关系组合推理研究[J]. 测绘科学技术学报, 2014, 31(5): 538-542.)
- [25] Cohen O D, Fibich G, Halperin D, et al. Conservative Visibility and Strong Occlusion for Viewspace Partitioning of Densely Occluded Scenes[J]. *Computer Graphics Forum*, 1998, 17(3): 243-253.
- [26] Naylor B, Amanatides J, Thibault W. Merging BSP Trees Yields Polyhedral Set Operations [J]. *ACM SIGGRAPH Computer Graphics*, 1990, 24(4): 115-124.
- [27] Huang Shan, Wang Botao, Wang Guoren, et al. A Survey on MapReduce Optimization Technologies [J]. *Journal of Frontiers of Computer Science & Technology*, 2013, 7(10): 885-905. (黄山, 王波涛, 王国仁, 等. MapReduce优化技术综述[J]. 计算机科学与探索, 2013, 7(10): 885-905.)
- [28] Wu Qunyong, Su Keyun, Zou Zhijie. A Map Reduce-Based Method for Parallel Calculation of Bus Passenger Origin and Destination from Massive Transit Data[J]. *Journal of Geo-Information Science*, 2018, 20(5): 647-655. (邬群勇, 苏克云, 邹智杰. 基于MapReduce的海量公交乘客OD并行推算方法[J]. 地球信息科学学报, 2018, 20(5): 647-655.)
- [29] Li Fan, He Honglin, Ren Xiaoli, et al. Research on Spatial Sensitivity Analysis Using Parallel Algorithm Based on MapReduce [J]. *Journal of Geo-Information Science*, 2014, 16(6): 874-881. (李帆, 何洪林, 任小丽, 等. 基于MapReduce的空间敏感性分析并行算法设计[J]. 地球信息科学学报, 2014, 16(6): 874-881.)
- [30] Lin Yaping, Du Zhenhong, Zhang Feng, et al. Research on the Analysis and Statistic of Geographical Conditions Based on the Strategy of "Grid Index+MapReduce" [J]. *Journal of Zhejiang University (Science Edition)*, 2017, 44(6): 660-665. (林雅萍, 杜震洪, 张丰, 等. "格网索引+MapReduce"策略下的地理国情统计分析研究[J]. 浙江大学学报(理学版), 2017, 44(6): 660-665.)