

doi: 10.6046/gtzyyg.2018.01.33

引用格式: 王彦佐,周伟,冯磊. 内存数据库在ZY1-02C海量数据空间检索中的应用[J]. 国土资源遥感,2018,30(1):238-242. (Wang Y Z, Zhou W, Feng L. Application of main memory database to spatial query of mass ZY1-02C data[J]. Remote Sensing for Land and Resources, 2018, 30(1): 238-242.)

# 内存数据库在ZY1-02C海量数据空间检索中的应用

王彦佐, 周伟, 冯磊

(中国国土资源航空物探遥感中心, 北京 100083)

**摘要:** 实现海量空间数据的高效检索是资源一号02C(ZY1-02C)卫星数据应用的重要前提。在传统基于关系型数据库的空间查询模式中, 频繁的磁盘I/O及内外存交换对查询性能有较大影响, 而完全基于内存的内存数据库技术可有效规避这一问题, 显著提升效率。基于Key-Value型内存数据库Redis, 研究并设计了矢量数据在Redis中的存储结构, 并实现了基于Key-Value型存储结构的R树索引。经实际应用检验, 该存储及索引架构能够有效提升海量空间数据的查询检索性能。

**关键词:** 内存数据库; 空间索引; 矢量数据

**中图法分类号:** TP 302 **文献标志码:** A **文章编号:** 1001-070X(2018)01-0238-05

## 0 引言

资源一号02C(ZY1-02C)卫星数据应用过程中, 为了保障矿产资源调查评价、地质灾害环境调查与监测等业务对数据时效性和准确性的要求, 需要针对所产生的数百万景原始影像及数据产品实现快速检索。空间数据检索操作的对象为矢量格式的空间分布图层。传统的空间数据检索架构大多基于关系型数据库(如Oracle)及空间数据引擎(如ArcSDE)<sup>[1]</sup>。该架构在I/O速率和存储结构等方面的限制对空间检索的效率有一定影响, 并且会随数据量增大而变得更显著。针对ZY1-02C海量空间数据检索的应用特点, 以及传统空间数据检索架构的限制, 采用了基于Key-Value型内存数据库的新型空间检索架构, 该架构的主要优势<sup>[2-3]</sup>有: ①读写速度快, 内存数据库中所有数据均存储于内存中, 其读写速度比存储于磁盘中的数据高出几个数量级, 可大大减少I/O操作的时间, 从而显著提升检索速度; ②复杂数据结构适应性强, 由于空间分布图层为矢量格式, 其数据结构比二维数据表复杂, 因此使用关系型数据库存储需要经过较复杂的数据类型转换, 而Key-Value型数据库架构灵活, 可以很好地适应复杂数据结构; ③受内存容量限制小, 内存数据库

可存储管理的数据量受内存容量限制, 远远小于关系型数据库的可用容量, 但由于ZY1-02C数据空间分布图层为矢量格式, 记录数多而单条记录占用量小, 总数据量有限(GB级), 不受该限制因素影响; ④性能的影响因素少, 关系型数据库对事务的要求非常严格, 在保证数据一致性的同时, 对数据读写效率也有一定影响; 在ZY1-02C数据空间检索中, 不存在复杂的多表间数据关系, 对数据原子性及一致性等要求也不高, 因此选用灵活性较强的内存数据库可以在一定程度上提高性能。依据该架构, 本文选用了互联网行业广泛使用的Redis数据库为基础平台, 研究并设计了矢量数据在Redis中的存储结构, 从而实现了基于Key-Value型存储结构的R树索引, 有效提升了ZY1-02C海量空间数据检索的性能。

## 1 Key-Value型内存数据库Redis

### 1.1 Key-Value型存储结构

Key-Value型存储结构<sup>[4]</sup>是一种存储管理数据的范式, 与之相对的是关系型存储结构, 它们之间有非常大的区别。

关系型存储结构中, 数据以“记录-字段”的形式存储于预定义好的二维数据结构中, 记录中每个

收稿日期: 2016-06-29; 修订日期: 2016-08-30

基金项目: 国土资源部公益性行业科研专项项目“资源一号02C星地质矿产与地质灾害环境应用模式及关键技术研究”(编号: 201311036)资助。

第一作者: 王彦佐(1986-), 男, 硕士, 主要从事地理信息系统方面的研究。Email: wangyanzuo@foxmail.com。

字段的类型都必须符合预定义的要求,且无法更改。

Key-Value 型存储结构则通过键-值对的形式存储数据,每一个唯一的键名(Key)对应一个唯一的值(Value),存取时通过键名来查找对应的值,值的类型和格式可任意定义,拥有较大的灵活性。同时,由于其灵活性较强,该结构占用的内存远小于关系型数据结构,因此内存数据库大多采用 Key-Value 型存储结构。

### 1.2 Redis 内存数据库

Redis<sup>[5]</sup>是一款开源的 Key-Value 型内存数据库,自发布以来已被 Twitter 及新浪微博等多家知名互联网公司使用,功能强大,运行稳定,本研究采用该数据库作为海量数据检索的基础平台。由于存储方式的巨大差异,传统关系型存储结构中的矢量数据存储方法及索引算法均无法应用于 Key-Value 型存储结构中,需要根据 Key-Value 型存储结构的组织形式进行重新设计及实现。

## 2 矢量数据在 Redis 中的存储结构

### 2.1 基于 Key-Value 型存储结构的矢量数据存储结构设计

对海量数据进行空间检索的前提是实现数据在 Redis 数据库中的存储,并能够对其进行高效地读写,结合 Key-Value 型存储结构以及 ZY1-02C 空间分布矢量数据特点,设计了用于存储矢量数据的 3 级结构<sup>[6]</sup>,分别为图层集(LayerSet)、图层(Layer)及要素(Feature)。

1) 图层集(LayerSet)。一个数据库中只有一个图层集对应的键-值对,键名即为“LayerSet”;键值使用集合形式,存储数据库中所有图层的图层名。

2) 图层(Layer)。一个数据库中可以有多个图层,每个图层的名称不可重复,并在图层集中登记;每个图层使用一个键-值对存储其元数据信息,键名即为图层名称,其格式为“Layer: [Name]”,其中 [Name] 可使用字符或数字的组合来区别不同的图层;键值使用 Hash 格式,存储图层的图层类型、图层范围、空间参考和空间索引名称等信息;图层及其包含的要素通过键名进行对应。

3) 要素(Feature)。一个图层可包含多个要素,每个要素使用一个键-值对存储其属性及空间信息;要素与其所在图层之间通过键名进行对应,要素的键名格式为“Layer: [Name]: Fea: [Index]”,其中前半部分即为所在图层的键名,后半部分为要素编号,其中 [Index] 以数字区分不同要素,不可重复;键值使用 Hash 格式,存储要素的属性及空间信

息;对于其中要素的属性信息,在 Hash 值中,子键名(SubKey)为字段名,子键值(SubValue)为字段值;对于其中要素的空间信息,在 Hash 值中,使用预定义的“Shape”作为子键名,将空间对象转换为符合 OGC 标准的 wkb 格式,以二进制的形式存储于子键值中。

### 2.2 存储结构示例

2.1 节中存储结构的示例如图 1 所示。

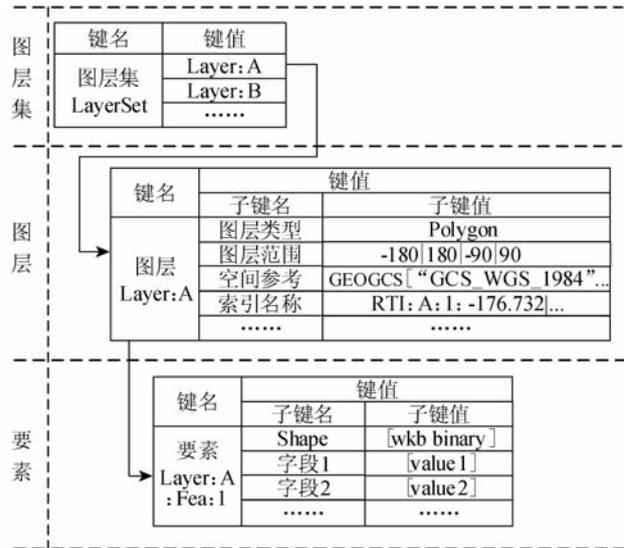


图 1 存储结构示例

Fig. 1 Storage structure example

## 3 基于 Key-Value 型存储结构的空間索引

### 3.1 空间索引基本概念

实现矢量数据在 Redis 数据库中的存储管理是进行空间检索操作的基础,为提高空间检索的效率,还需要为数据库中的矢量数据建立空间索引。

空间索引<sup>[7-8]</sup>是一种数据结构,使用预定义的算法将数据库中各要素的空间特征(外接矩形和中心点坐标等)与其存储地址之间的对应关系组织并保存起来。通过空间索引,用户在空间检索时不需要对全库进行遍历,即可快速定位到所需的要素,从而大大提高检索效率。

传统的空间索引大多基于文件或关系型数据库,Key-Value 型内存数据库由于其数据存储结构的差异,无法直接使用传统的空间索引存储结构,需要依据基础索引算法,设计并实现有针对性的空间索引。

### 3.2 R 树空间索引基础算法

常见的空间索引算法包括格网索引、空间哈希索引和 R 树索引等,其中 R 树索引<sup>[9-10]</sup>是目前空间

索引领域应用最广泛的索引算法,其特点为查询维护便捷,效率高。本研究选用 R 树索引作为生成空间索引的基础算法。

R 树索引是 B 树索引向二维空间的扩展,其基础算法为: ①R 树中的所有数据都存储于叶节点中,其他非叶节点以最小外包矩形(minimum bounding rectangle, MBR)的格式存储辅助数据; ②每个非叶节点的 MBR 范围为其所有子节点 MBR 范围的 MBR; ③R 树生成过程中,同一层的各节点之间,矩形相交部分应尽量小; 位置上相近的结点尽量在树中聚集为一个父节点。

### 3.3 R 树索引在 Redis 数据库下的存储结构

基于 R 树索引基础算法,设计并实现了基于 Key - Value 型存储结构的 R 树索引存储结构及检索方案,其中存储结构详情如下:

- 1) R 树索引在 Redis 数据库中均采用集合格式存储。
- 2) 每个图层对应一组索引,该组索引由一个根

节点、多个中间节点及叶节点构成,图层的元数据信息中记录根节点的键名,以标识图层与索引之间的对应关系。

3) 一组索引由数据库中的多个键 - 值对描述,每一个节点对应一个键 - 值对,其键名为当前节点的信息,键值为集合格式,记录其包含的所有下一级子节点或要素的键名。

4) 各个节点主要信息均记录在键名中,即

$$RTI: \#: * : [WestBound] | [EastBound] | [SouthBound] | [NorthBound] \quad (1)$$

其中: #为索引序列号,按索引编制顺序顺次编码,对应某一图层的一组索引序列号相同; \* 为索引层号,根节点层号为 1,其子节点层号为 2,依此类推; [ WestBound ], [ EastBound ], [ SouthBound ] 和 [ NorthBound ] 分别为该节点对应 MBR 的西、东、南和北边界。

该索引存储结构示例如图 2 所示。

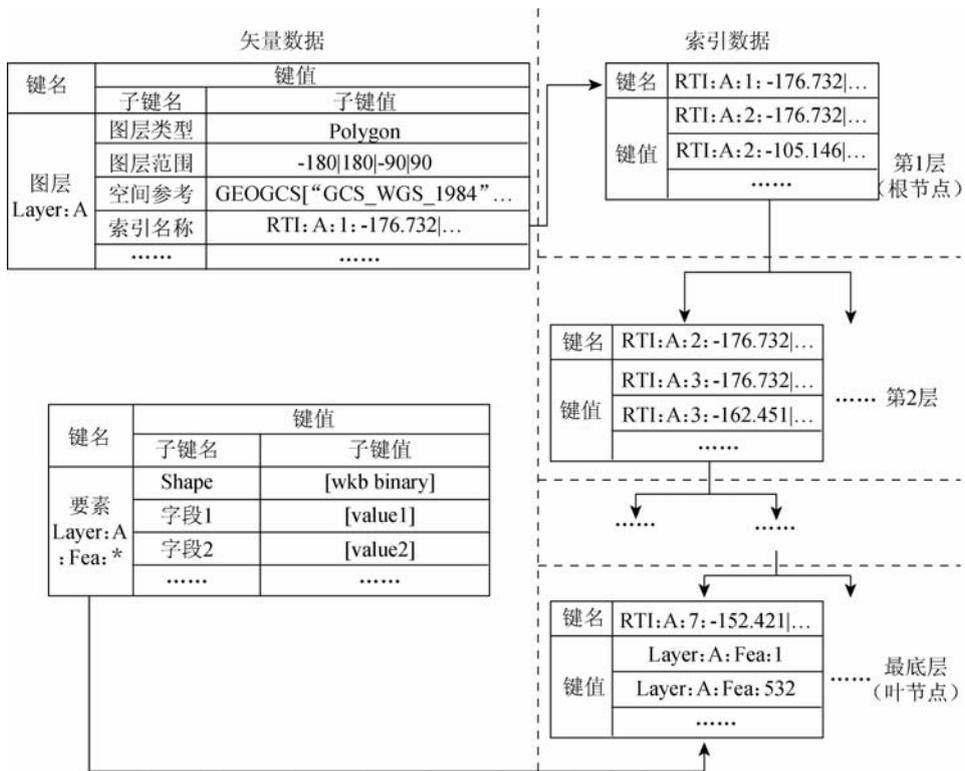


图 2 R 树存储结构示例

Fig. 2 Storage structure example of R - tree index in Redis

## 4 系统实现

### 4.1 系统基础环境

系统实现过程中,选取的计算机软硬件基础环境如下: ①操作系统选用 Windows Server 2008R2; ②内存数据库软件选用 Redis 2. 8 On Windows;

③集成开发环境选用 Microsoft Visual Studio. net 2010; ④开发语言选用 C#; ⑤GIS 组件选用开源栅格空间数据转换库(geospatial data abstraction library, GDAL)/OGR 库。

### 4.2 矢量数据转换入库

该功能将 Shapefile 格式的矢量数据转换为 2. 1 节中的存储结构,并存入 Redis 数据库,其实现流程

如图 3 所示。

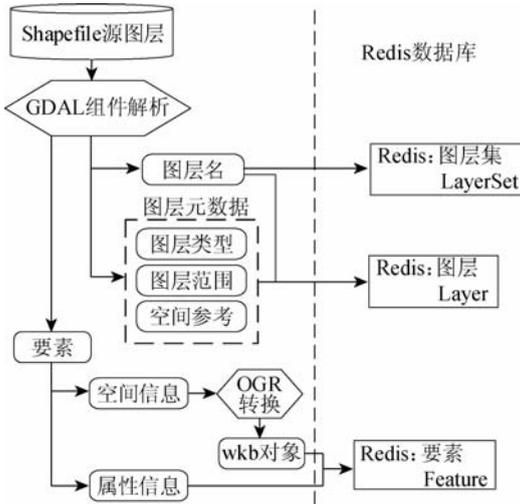


图 3 矢量数据转换入库流程

Fig. 3 Data conversion flowchart of shapefile

图 3 流程中,首先使用 GDAL 组件对图层文件进行解析,获取图层名及元数据信息,并将上述信息存入 Redis 数据库中的“图层集”以及“图层”键-值对;同时逐个读取矢量要素,并使用 GDAL 组件将空间信息转为 wkb 格式,与属性信息一同存入 Redis 中的“要素”键-值对。

### 4.3 Key-Value 型结构下 R 树索引空间检索

该功能根据输入的空间查询条件(包含目标图层及查询空间对象),依据所构建的 R 树索引,查找并返回数据库中符合空间条件的矢量要素,其流程如图 4 所示。

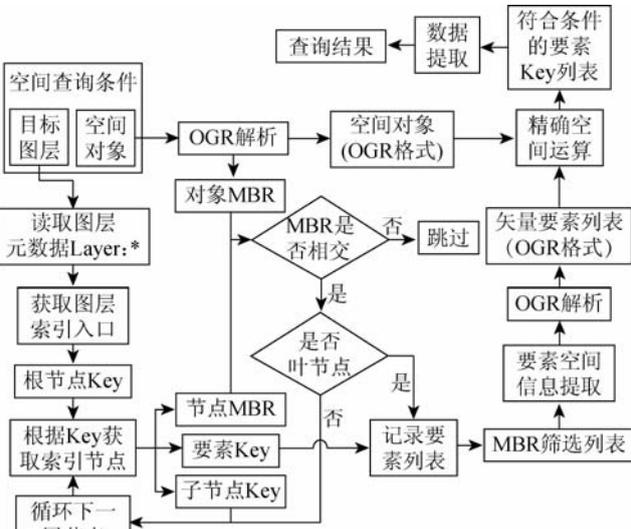


图 4 空间数据检索流程

Fig. 4 Flowchart of spatial data query

### 4.4 空间检索效率测试与对比

系统功能实现后,在相同软硬件环境下,对比了基于 Redis 内存数据库的 R 树索引以及传统基于 Oracle 数据库的格网索引 2 种空间检索架构的检索

效率。

测试软硬件环境为:VMWare 虚拟机,二路二核 CPU,8 GB 内存,Redis2. 8. 19 和 Oracle11. 1. 0. 6 软件。测试数据为:ZY1-02C 原始影像空间分布图层,共包含 411 523 个要素。矩形空间范围要素检索测试结果如图 5 所示。

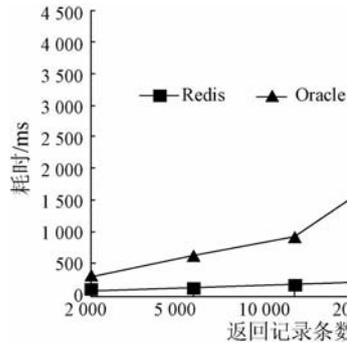


图 5 空间数据检索效率测试

Fig. 5 Efficiency test of spatial query

从对比测试结果中可以看到,基于 Key-Value 型内存数据库 Redis 进行空间检索操作,能够在海量数据条件下,有效提高空间检索的效率。

## 5 结论

本文针对 ZY1-02C 海量空间数据检索的应用需求,以及传统空间检索架构的限制,基于 Key-Value 型内存数据库技术,设计并实现了一种矢量空间数据的存储架构以及空间索引架构,完成了相应功能开发,现已部署于 ZY1-02C 相关应用系统中。经对比测试及实际运行检验,该架构与传统基于磁盘的关系型数据库架构相比,能够更好地适用于 ZY1-02C 海量空间数据检索的应用场景,并有效提升空间检索操作的效率,取得了良好的应用效果。

### 参考文献 (References):

[1] 薛涛,刁明光,李建存,等. 资源环境遥感海量空间数据存储、检索和访问方法[J]. 国土资源遥感,2013,25(2):168-173. doi:10.6046/gtzyyg.2013.02.28.  
Xue T, Diao M G, Li J C, et al. Approach to storing, retrieving and accessing mass spatial data in resources and environments remote sensing[J]. Remote Sensing for Land and Resources, 2013, 25(2):168-173. doi:10.6046/gtzyyg.2013.02.28.

[2] 杨武军,张继荣,屈军锁. 内存数据库技术综述[J]. 西安邮电学院学报,2005,10(3):95-99.  
Yang W J, Zhang J R, Qu J S. An overview of main-memory database technologies[J]. Journal of Xi'an University of Post and Telecommunications, 2005, 10(3):95-99.

[3] 王珊,肖艳芹,刘大为,等. 内存数据库关键技术研究[J]. 计算机应用,2007,27(10):2353-2357.

- Wang S, Xiao Y Q, Liu D W, et al. Research of main memory database[J]. Computer Applications, 2007, 27(10): 2353 - 2357.
- [4] Wikipedia. Key - value database[EB/OL]. (2016 - 05 - 16) [2016 - 06 - 27]. [https://en.wikipedia.org/wiki/Key-value\\_database](https://en.wikipedia.org/wiki/Key-value_database).
- [5] Redislabs. Redis[EB/OL]. (2016 - 06 - 17) [2016 - 06 - 27]. <http://redis.io>.
- [6] 朱进, 胡斌, 邵华, 等. 基于内存数据库 Redis 的轻量级矢量地理数据组织[J]. 地理信息科学, 2014, 16(2): 165 - 172.
- Zhu J, Hu B, Shao H, et al. Research of lightweight vector geographic data management based on main memory database Redis[J]. Journal of Geo - information Science, 2014, 16(2): 165 - 172.
- [7] 阎超德, 赵学胜. GIS 空间索引方法述评[J]. 地理与地理信息科学, 2004, 20(4): 23 - 26, 39.
- Yan C D, Zhao X S. The review of spatial indexes in GIS[J]. Geography and Geo - Information Science, 2004, 20(4): 23 - 26, 39.
- [8] 史文中, 郭薇, 彭奕彰. 一种面向地理信息系统的空间索引方法[J]. 测绘学报, 2001, 30(2): 156 - 161.
- Shi W Z, Guo W, Peng Y Z. A spatial indexing method for GIS[J]. Acta Geodaetica et Cartographica Sinica, 2001, 30(2): 156 - 161.
- [9] 黄晓明, 杨红雨, 闫觅. 基于 R 树的空管 GIS 数据模型索引的设计[J]. 微计算机信息, 2008, 24(8-3): 144 - 146.
- Huang X M, Yang H Y, Yan M. The index design for ATC GIS data model based on R tree[J]. Microcomputer Information, 2008, 24(8-3): 144 - 146.
- [10] 罗琪, 李军, 陈萃. 基于 GIS 平台的 R 树索引模型研究与实现[J]. 计算机工程与科学, 2003, 25(6): 93 - 96.
- Luo Q, Li J, Chen L. Research and implementation of a R - tree index model based on the GIS platform[J]. Computer Engineering and Science, 2003, 25(6): 93 - 96.

## Application of main memory database to spatial query of mass ZY1 - 02C data

WANG Yanzuo, ZHOU Wei, FENG Lei

(China Aero Geophysical Survey and Remote Sensing Center for Land and Resources, Beijing 100083, China)

**Abstract:** Fast spatial query is the foundation of the mass ZY1 - 02C data application. In the traditional spatial query structure using relational database, frequent disk I/Os and data swaps degrade the query performance a lot. In contrast, main memory database which is completely based on the memory can significantly improve the searching efficiency. In this study, a vector data storage and R - tree indexing structure was designed and implemented based on the Redis database, which is a main memory database of Key - Value type. This structure was used in practice and the efficiency of spatial query was improved.

**Keywords:** main memory database; spatial index; vector data

(责任编辑: 陈理)