

# 面向分布式 GIS 空间数据的 Key-value 缓存

涂振发<sup>1</sup> 孟令奎<sup>1</sup> 张 文<sup>1</sup> 黄长青<sup>1</sup>

(1 武汉大学遥感信息工程学院,武汉市珞喻路 129 号,430079)

**摘 要:**设计了一种基于 Key-value 结构的缓存 KV-cache,旨在简化数据结构,高效管理缓存数据。KV-cache 由 Mem-cache 和 F-cache 构成。其中,Mem-cache 为第一级缓存,并采用缓存回写机制以异步方式将数据写入 F-cache 进行持久化存储。F-cache 使用大文件存储数据,并构建了基于 B 树的缓存索引以加快检索速度。数据写入 F-cache 的位置由基于 B 树的空闲空间管理器进行分配。在 100 Mbps 局域网内测试 KV-cache 读写性能。实验表明,KV-cache 读写性能优于传统文件目录方式的缓存,能够安全、高效地管理所缓存的各种空间数据。

**关键词:**分布式 GIS;缓存;KV-cache;Key-value 存储;空闲空间管理

**中图法分类号:**P208; TP311.13

随着网络技术的发展,分布式 GIS 在越来越多的领域得到应用,并取得了良好的效果。分布式 GIS 所涉及的空间数据来源广泛,种类繁多,结构复杂,数据量极大,具有明显的空间特征<sup>[1-2]</sup>。海量空间数据存取代价和网络传输代价是影响分布式 GIS 性能的重要因素<sup>[3-5]</sup>。大量用户的并发请求将给服务器端极大的压力,数据的传输效率也在很大程度上影响性能。缓存机制是解决上述问题,提高分布式 GIS 性能的一种有效手段。

缓存机制将需要频繁访问的数据保存在本地,以减少数据的请求频率和网络传输量,从而提高性能。许多学者针对此问题进行了研究,并提出了各自的解决方法<sup>[3-9]</sup>。NASA 的 World Wind 则采用基于文件目录方式构建客户端缓存,以避免频繁请求数据<sup>[4, 10]</sup>。这种基于文件目录构建缓存的方式虽简便易行,却存在数据保密性差,空间利用率低,检索效率低下,硬盘 I/O 频繁等缺点<sup>[4]</sup>。为此本文尝试同时使用内存和单个大文件构建一种基于 Key-value 结构的缓存,旨在简化数据结构,高效管理所缓存的各类空间数据。

## 1 Key-value 存储

Key-value 存储是一种非常简洁的数据存储

方式,由关键字(Key)和值(Value)构成。其中,Value 存储实际数据,这些数据包括字符串、整数或二进制数据等;Key 则用来标识数据,通常使用字符串或整数。Key-value 存储具有模式自由,查询速度快,并发高等优点,缺点是不支持复杂的查询<sup>[11-12]</sup>。目前,Key-value 存储已广泛应用于数据存储、缓存等领域,特别是随着云计算的发展,大量分布式 Key-value 结构的存储系统不断推出,较为著名的有 Bigtable<sup>[12]</sup>, Dynamo<sup>[11]</sup>, Cassandra<sup>[13]</sup>等。因此,本文尝试使用 Key-value 结构构建面向分布式 GIS 的海量空间数据缓存系统。

## 2 缓存设计

### 2.1 缓存架构

本文所述缓存(KV-cache)是一种基于 Key-value 结构的双缓存系统,它被设计用于分布式 GIS 的客户端以缓存遥感影像、矢量数据、DEM 等类型的空间数据。KV-cache 包括内存缓存(Mem-cache)和文件缓存(F-cache)。其中,Mem-cache 由于访问速度远高于 F-cache,位于 F-cache 之上,为第一级缓存;F-cache 采用单个大文件方式构建,包括存储数据的数据文件和存储索引的

收稿日期:2013-07-08。

项目来源:水利部公益性行业科研专项经费资助项目(201001046);国家科技支撑计划资助项目(2011BAH16B08)。

索引文件,为第二级缓存。

图1为KV-cache的工作流程。

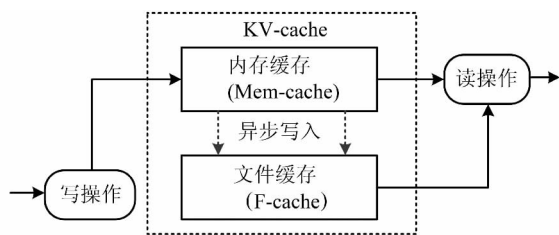


图1 KV-cache的工作流程

Fig.1 Workflow of KV-cache

### 2.2 缓存块及其索引

缓存块是整个缓存的最基本单元,它采用 Key-value 结构构建。其中,Key 可使用字符串或整数,为缓存块唯一标识,其值由用户按一定算法计算得到。虽然不同缓存中 Key 的数据类型和长度均是可变的,但同一缓存内必须始终一致。Value 用来存储数据,各类数据均以字节数组的形式存储在 Value 内。为保证读写效率,Key 和 Value 最大长度分别限定为 256 B 和 100 MB。

每个缓存块均对应一个缓存块索引。缓存块索引同样采用 Key-value 结构,并使用相同的 Key 值与缓存块关联。缓存块索引的 Value 采用可扩展结构来存储缓存块描述信息,其基本结构如图2所示。图2为使用缓存块及其索引存储影像金字塔瓦片数据的示例。其中,偏移地址和数据大小均为 64 位整数,用来定位 F-cache 中的缓存块;数据类型、压缩方式、加密方式均使用 8 位无符号整数表示,用来说明所存储的空间数据为何种类型,采用何种压缩和加密方式;存储时间为缓存块写入时间,使用 64 位整数记录;地理范围用来描述空间数据的地理范围。缓存块索引 Value 的结构可根据应用目的的不同进行扩展,但在同一缓存中其结构一旦确定必须始终不变。由于 B 树已广泛应用于 Oracle、MySQL 等数据库系统,并且其可减少定位记录所经历的中间过程,加快存取速度,因此本文使用 B 树构建缓存索引。

### 2.3 内存缓存 (Mem-cache)

Mem-cache 构建于 F-cache 之上,作为 F-cache 的缓冲,以减少硬盘 I/O 次数,提高响应速度。Mem-cache 同样使用 B 树组织缓存块,使用 Key 值查找所需数据。Mem-cache 初始化时并不立即分配容量,而仅指定其所能使用的最大容量。

Mem-cache 中的数据使用缓存回写机制写入 F-cache 进行持久化存储。执行缓存回写的条

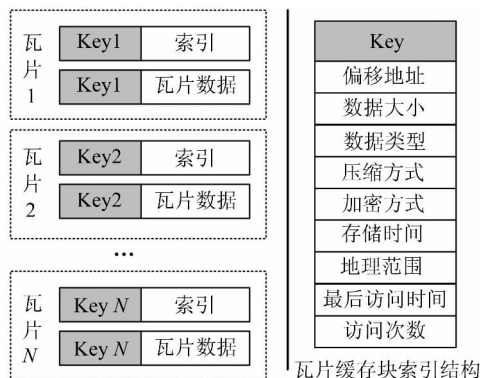


图2 缓存影像金字塔瓦片示例

Fig.2 A Sample of Caching Image Pyramid

件包括定时时间到达和存储容量达到阈值。定时时间间隔设置不宜过大,否则容易造成数据丢失;也不宜过小,否则增加硬盘读写次数。当容量达到阈值时,Mem-cache 将使用先进先出算法(FIFO)执行缓存置换。为了减少缓存置换的执行次数,特别是数据频繁写入时,过多缓存置换造成的性能浪费,Mem-cache 每次均剔除指定比例(如最大容量 20%)的缓存块。缓存块在剔除前将写入 F-cache 进行持久化存储。

### 2.4 文件缓存 (F-cache)

F-cache 采用大文件方式构建,由数据文件和索引文件构成,如图3所示。

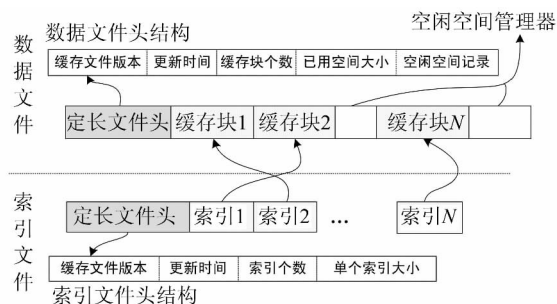


图3 F-cache存储结构示意图

Fig.3 Schematic Diagram of F-cache Structure

数据文件和索引文件均由文件头和数据存储区域两部分构成。数据文件使用单个大文件构建,可支持几 GB 甚至几十 GB 大小的文件,其大小在初始化时分配,并保持不变。所有数据均存储在数据存储区域。写入时首先将缓存块序列化为字节数组,然后整体、连续地写入空闲空间内,与此同时更新对应索引的偏移地址和数据大小。读取缓存块只需要一次硬盘搜索即可完成:首先使用 Key 查找索引以获取缓存块的存储地址和大小,然后直接使用此地址和大小读取数据。

索引文件使用一个小文件创建,其大小依据缓存块索引个数动态调整。所有缓存块索引的大

小均一致,并采用顺序方式存储。索引的个数和单个索引大小均保存在定长的索引文件头内。F-cache 在初始化时读取所有缓存块索引,并构建缓存索引。缓存索引在整个运行期间一直驻留在内存中,以定时方式写回索引文件。

同一时间内,F-cache 仅允许一个写操作进行,但多个读操作可并发执行。

图 3 中,数据文件空闲空间由空闲空间管理器进行管理。单个空闲空间由偏移地址和空间大小表示,二者均为 64 位整数。空闲空间管理器基于 B 树构建。每次分配存储空间均分配大小最为合适的空闲空间。若包含多个大小最合适的空闲空间,则偏移地址最小的将被分配。

如图 4 所示,数据写入 F-cache 时,若仅包含一个连续空闲空间,则将数据写入该空闲空间的起始位置;若存在多个连续空闲空间,数据将被写入大小最为合适的空闲空间内;若所有连续空闲空间均不足,则首先移除部分数据以释放出充足连续空闲空间,然后将数据写入此空闲空间。

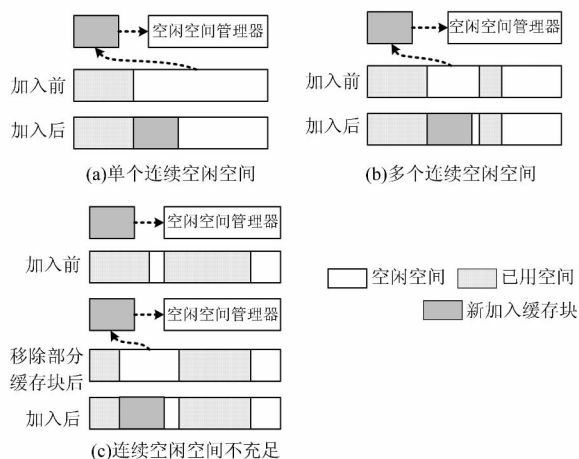


图 4 几种情况下 F-cache 写入缓存块示意图  
Fig. 4 Schematic Diagram of Writing Cache Block to F-cache Under Several Conditions

当使用 F-cache 缓存瓦片金字塔时,每个瓦片均被视为一个缓存块,其索引结构如图 2 所示。

### 3 实验与结果分析

整个缓存使用 C# 实现,并在 100 Mbps 局域网内进行实验,测试缓存读写性能。实验系统基于 C/S 结构构建,使用日志记录实验结果。服务器端配置为:CPU 为 AMD Phenom II X4 955,4 GB 内存,1 TB 硬盘,操作系统为 Windows Server 2003;客户端配置为:CPU 为 Intel Dual E2200,2 GB 内存,500 GB 硬盘,操作系统为

Windows XP。实验数据使用全球 90 m 分辨率的 SRTM 数据构建的影像金字塔,共构建 10 级金字塔,无数据部分以补零方式补齐。金字塔瓦片大小为 512 像素×512 像素,以 jpg 格式存储,共生成瓦片 700 975 个,总大小约为 8.57 GB,单个瓦片大小为 5~154 KB。图 5 为客户端渲染 SRTM 影像金字塔示例。

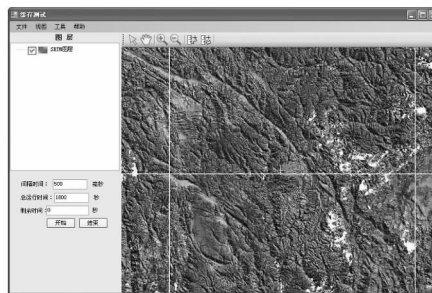


图 5 客户端渲染 SRTM 影像金字塔  
Fig. 5 Rendering SRTM Image Pyramid in the Client

实验中使用 C# 实现了一个基于文件目录方式缓存(简称文件目录缓存),并将其与 KV-cache、F-cache 进行对比,测试各个缓存的读写性能。各个缓存的最大容量均为 1 GB,定时保存间隔为 5 min,KV-cache 中 Mem-cache 最大容量为 50 MB。实验结果如图 6 所示,KV-cache、F-cache 的读写性能均优于文件目录缓存。图 6(a)中,由于需执行定时保存,KV-cache、F-cache 在第 5 min、10 min、15 min、20 min、25 min、30 min 写入性能有较大波动,但随着运行时间的增长,波动幅度逐渐减少,并最终趋于稳定。由于使用了 Mem-cache,KV-cache 减少了硬盘写入次数,其写入性能较 F-cache 有一定提升。如图 6(b)所示,由于起始阶段需进行初始化,KV-cache 的读取性能较差;使用 Mem-cache 节省了大量从硬盘读取数据和将数据反序列化的时间,KV-cache 的读取性能远优于另外两种缓存。即使单独使用 F-cache 作为缓存,其读取性能也优于文件目录缓存。

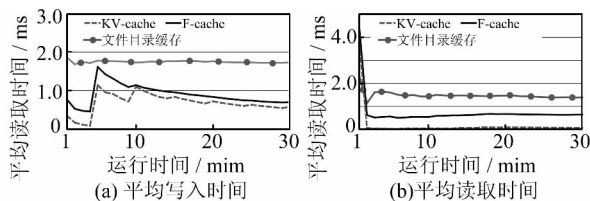


图 6 不同缓存平均读写时间  
Fig. 6 Average Read/Write Time of Different Caches

图 7 为在不同定时保存时间间隔下 KV-cache 的读写性能测试。如图 7(a)所示,在 Mem-cache 最大容量相同的情况下,缓存写入性能的

波动更加频繁。这是由于缩短定时保存间隔后, KV-cache 需执行更多的保存操作。在前 15 min, 定时 5 min 情况下, KV-cache 的写入性能较定时 2 min 占优; 在后 15 min, 前者则略逊于后者, 但二者差距并不大。因此, 增大定时保存时间对提高 KV-cache 的写入性能作用不大。如图 7(b) 所示, 在 Mem-cache 最大容量相同的情况下, 增长定时保存时间间隔可减少并发读取的同步锁定, 从而提高读取性能, 但优势并不明显。

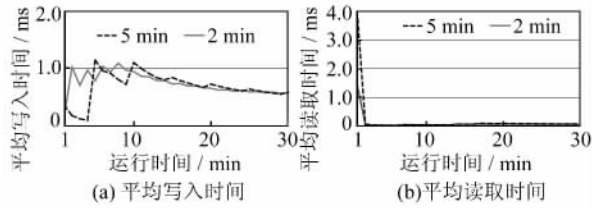


图 7 不同定时保存时间下 KV-cache 平均读写时间  
Fig. 7 Average Read/Write Elapsed Time of KV-cache with Different Timing Intervals

图 8 为 KV-cache 在不同 Mem-cache 最大容量下的读写性能测试。如图 8(a) 所示, 在定时 5 min 的情况下, 具有较大容量的 Mem-cache 可存储更多的数据, 定时保存时需写入更多的数据。因此, 初始阶段, 增加 Mem-cache 的最大容量反而会略微降低写入性能。但随着运行时间的增长, 由于大部分数据已写入 F-cache, 容量大小对于写入性能的影响已不明显。如图 8(b) 所示, 由于较多数据可直接从内存中读取, 增大 Mem-cache 容量有助于提高 KV-cache 的读取性能。

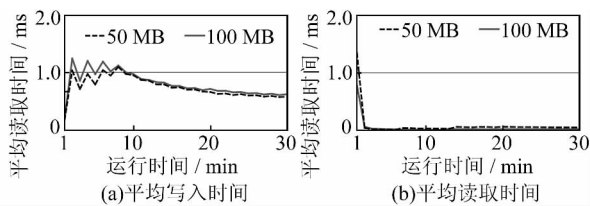


图 8 不同 Mem-cache 容量下 KV-cache 平均读写时间

Fig. 8 Average Read/Write Time of KV-cache in Different Mem-cache Capacities

## 4 结 语

基于 DPGGrid 的数据管理系统是一个 C/S 结构、面向 DPGGrid 所生产海量空间数据的管理系统, 涉及遥感影像、影像金字塔、矢量数据和三维数据模型等。目前, KV-cache 已应用于该系统, 用于数据浏览客户端各类空间数据的缓存。实践表明, KV-cache 读写性能较传统的基于文件目录

方式的缓存有较大的提升, 且能够安全、高效地管理缓存内的各种空间数据。但是, 目前 KV-cache 仅被用于客户端缓存, 下一步的工作是将其扩充为分布式缓存, 用于分布式 GIS 的服务器端。

## 参 考 文 献

- [1] 孟令奎, 史文中, 张鹏林, 等. 网络地理信息系统原理与技术[M]. 2 版. 北京: 科学出版社, 2010
- [2] 吴华意, 章汉武. 地理信息服务质量(QoGIS): 概念和研究框架[J]. 武汉大学学报·信息科学版, 2007, 32(5): 385-388
- [3] 祁羽, 陈萃, 张瑞雪, 等. 基于双缓存机制的分布式 WebGIS 数据集成访问策略[J]. 计算机工程与科学, 2007, 5: 41-44
- [4] 熊汉江, 杨哲宇, 戴雪峰, 等. 网络 3 维地球软件客户端缓存动态管理设计与实现[J]. 地理信息世界, 2009, 4: 55-59
- [5] 李浩松, 朱欣焰, 李京伟, 等. WebGIS 空间数据分布式缓存技术研究[J]. 武汉大学学报·信息科学版, 2005, 30(12): 1 092-1 095
- [6] 曾武, 王浩, 徐正全, 等. 网络地理信息系统中的栅格数据集缓存[J]. 华中科技大学学报(自然科学版), 2009, 9: 37-39
- [7] Yang C W, Wong W D, Yang R X, et al. Performance-improving Techniques in Web-based GIS[J]. International Journal of Geographical Information Science, 2005, 19(3): 319-342
- [8] 潘少明, 喻占武, 李锐. 基于主动缓存的 P2P 海量地形漫游瓦片调度算法[J]. 测绘学报, 2009, 3: 236-241
- [9] 罗英伟, 汪小林, 许卓群. 层次化 WebGIS 构件系统的设计与实现[J]. 计算机学报, 2004, 2: 177-185
- [10] NASA. NASA World Wind[OL]. <http://worldwind.arc.nasa.gov/index.html>, 2011
- [11] Decandia G, Hastorun D, Jampani M, et al. Dynamo: Amazon's Highly Available Key-value Store[J]. ACM SIGOPS Operating Systems Review, 2007, 41(6): 205-220
- [12] Chang F, Dean J, Ghemawat S, et al. Bigtable: A Distributed Storage System for Structured Data[J]. ACM Transactions on Computer Systems (TOCS), 2008, 26(2): 1-26
- [13] Lakshman A, Malik P. Cassandra - A Decentralized Structured Storage System[J]. ACM SIGOPS Operating Systems Review, 2010, 44(2): 35-40

第一作者简介: 涂振发, 博士生, 研究方向为空间数据存储和管理、云 GIS 等。

E-mail: tuzhenfa@126.com

## Key-value Caching of Geospatial Data for Distributed GIS

TU Zhenfa<sup>1</sup> MENG Lingkui<sup>1</sup> ZHANG Wen<sup>1</sup> HUANG Changqing<sup>1</sup>

(1 School of Remote Sensing and Information Engineering, Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

**Abstract:** The access and network transmission costs for geospatial data in distributed GIS are important factors effecting performance. The cache is one of the effective measures to solve performance problems. In this paper, we designed a cache named KV-cache based on Key-value structure to manage geospatial data efficiently with a simple data structure. A KV-cache consists of a memory cache called Mem-cache and a file cache called F-cache. Mem-cache is the first level cache, the cache write-back mechanism guarantees that data is being written to the F-cache asynchronously. The F-cache is created by a single large file, while a cache indexer based on B-tree, is built to speed up the query. A free space manager which is also based on B-tree is used to allocate the data storage address for F-cache. Experiments compare the read/write performance of KV-cache with the traditional file cache in the 100 Mbps LAN. The experiments show that the read/write performance of KV-cache is much better than file cache, thus managing geospatial data more securely and efficiently.

**Key words:** distributed GIS; cache; KV-cache; Key-value storage; free space manager

---

**About the first author:** TU Zhenfa, Ph D candidate, specialized in spatial data storage and management, cloud GIS, etc.

E-mail: tuzhenfa@126.com

.....  
(上接第 1338 页)

## Local Terrain Deformation in Panoramic Maps

LI Lin<sup>1</sup> YU Zhonghai<sup>1</sup> LI Zhen<sup>1</sup> XIANG Zhujun<sup>2</sup>

(1 School of Resource and Environmental Sciences, Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

(2 Troops 61363, Xi'an 710054, China)

**Abstract:** Panoramic maps are usually used to show the landscape from a specific direction and viewing height in a central perspective projection. Regardless of the direction and height, some landscape features may be obscured. In order to improve the visual effect of panoramic maps, a method for local terrain deformation in the horizontal direction based on liquefaction deformation is proposed. This method for reducing local terrain elevation is based on eye tracking, a method that raises the local terrain elevation using a Gaussian transform. This method reduces occlusion and shrinkage phenomenon, enhancing the artistic visual effects of a panoramic map.

**Key words:** panoramic maps; local terrain distortion; liquefaction deformation; Gaussian transform; gaze tracking

---

**About the first author:** LI Lin, professor, Ph D supervisor. He is engaged in cartography, geo-ontology and 3D cadaster.

E-mail: lilin@whu.edu.cn