

基于多尺度复合金字塔模型的缓存策略研究

王 贺¹, 王志宝¹, 陈良富², 赵 亮¹

(1. 东北石油大学 计算机与信息技术学院, 黑龙江 大庆 163318;

2. 中国科学院空天信息创新研究院遥感科学国家重点实验室, 北京 100000)

摘 要:为了解决多源异构数据具有的统一组织和集成管理问题,提出了一种基于多尺度复合金字塔模型的数据组织方法。同时,针对海量瓦片数据传输给服务器和网络带来的压力过大、响应时间过长的问题,通过在客户端建立缓存机制和瓦片缓存索引,提出了一种基于多尺度复合金字塔模型的瓦片数据缓存替换算法 MCPCR。该算法在传统缓存替换算法的基础上,适用于加载多类型数据,并综合考虑了用户操作习惯以及引入了瓦片保护机制。以北京市大气污染 PM2.5 数据、风场数据及影像数据为例,与传统缓存替换算法进行对比。实验结果证明,多尺度复合金字塔模型能够实现对瓦片数据的统一组织和集成管理,基于多尺度复合金字塔模型的瓦片数据缓存替换算法能够对多种类型数据进行加载,相对于传统缓存替换算法可以提高瓦片命中率及字节命中率,提高用户响应速度。

关键词:缓存策略;多尺度;复合金字塔模型;数据组织;瓦片数据

中图分类号:TP333

文献标识码:A

文章编号:1673-629X(2022)02-0101-06

doi:10.3969/j.issn.1673-629X.2022.02.016

Research on Caching Strategy Based on Multi-scale Compound Pyramid Model

WANG He¹, WANG Zhi-bao¹, CHEN Liang-fu², ZHAO Liang¹

(1. School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China;

2. State Key Laboratory of Remote Sensing Science, Aerospace Information Research Institute of Chinese Academy of Sciences, Beijing 100000, China)

Abstract: In order to solve the problem of unified organization and integrated management of multi-source heterogeneous data, we propose a data organization method based on multi-scale compound pyramid model. At the same time, in view of the excessive pressure and long response time caused by the transmission of massive tile data on the server and the network, by establishing a caching mechanism and tile cache index on the client, we propose a tile data cache replacement algorithm MCPCR based on multi-scale compound pyramid model. Based on the traditional cache replacement algorithm, the proposed algorithm is suitable for loading multiple types of data, and comprehensively takes the user's operating habits into account and introduces a tile protection mechanism. Taking Beijing's PM2.5 data, wind field data and image data as examples, the proposed algorithm is compared with the traditional cache replacement algorithm. The experiment proves that the multi-scale composite pyramid model can realize the unified organization and integrated management of tile data. The MCPCR algorithm can load multiple types of data, increase the tile hit rate and byte hit rate compared with traditional cache replacement algorithm, and improve user response speed.

Key words: caching strategy; multi-scale; compound pyramid model; data organization; tile data

0 引言

网络地理信息系统(WebGIS)是网络和地理信息系统的结合,以互联网协议和终端为基础形成的客户端地理信息应用系统^[1],与人们的日常生活密不可分。传统的WebGIS系统通过客户端向服务器发送地理信息相关的请求,服务器收到后将相应的数据返回给客

户端^[2]。该方法受网络速度和服务器配置的影响,存在响应时间长、渲染效率低的缺点^[3]。为了提高客户端显示效率,构建金字塔模型是重要的解决方案^[4],在不影响画面视觉效果的同时,构建不同分辨率的多级数据来提高绘制速度,节省绘制所需时间^[5]。构建金字塔模型的关键环节是对空间数据进行组织,经过组

收稿日期:2021-02-23

修回日期:2021-06-23

基金项目:国家重点研发计划(2017YFB0503901)

作者简介:王 贺(1994-),男,硕士研究生,研究方向为大数据可视化;王志宝,博士,副教授,研究方向为大气遥感研究。

织后的瓦片数据可以提升检索效率^[6]。

在大数据时代,由于瓦片数据的海量特性和用户规模的不断增长,仍然存在网络拥塞、服务器过载等问题^[7]。通过研究瓦片缓存策略,以读取缓存数据的方式来减少对服务器的请求数量,可以在数据组织的基础上减轻服务器和网络传输的压力^[8]。因此,改进现有的瓦片缓存策略对网络地理信息系统具有重要意义。

在瓦片金字塔的基础上,该文提出了一种多尺度复合金字塔模型。与传统的瓦片金字塔模型相比,它可以统一组织和管理多种类型数据。并在此基础上提出了一种基于多尺度复合金字塔模型的缓存置换策略(multi-scale compound pyramid cache replacement),相较于现有的瓦片缓存算法,进行了如下优化:(1)支持

加载多种类型数据;(2)考虑瓦片数据的空间特性,根据用户操作习惯,动态计算当前命中瓦片相邻及相邻层级瓦片数据的请求频次;(3)引入瓦片保护机制。在瓦片进入到缓存空间后,一段时间内无法被置换。这种方式可以有效解决当缓存空间已满时,新进入到缓存空间的瓦片数据由于价值最低而被置换的问题。

1 构建多尺度复合金字塔模型

对多种类型数据进行组织,传统的做法是将多种类型数据单独建立索引,建立不同的金字塔模型以分开管理^[9]。采用这种方式金字塔的构建效率低、管理难度大。所以该文提出了一种能够统一管理各类数据的多尺度复合金字塔模型,如图 1 所示。

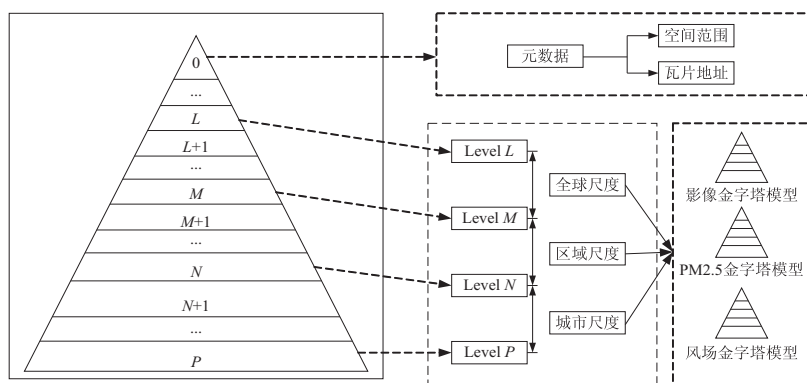


图 1 多尺度复合金字塔模型

多尺度复合金字塔模型从金字塔的底层到顶层,分辨率逐渐降低,包含了全球、区域、城市三个尺度数据,在不同尺度下又由多种类型数据共同组成,且多种类型数据在相同层级下表示的地理范围一致。

在多尺度复合金字塔模型中,在每种尺度下分别定义 1 个层级作为实际物理储存层级,负责储存该尺度范围内的所有信息。如图 2 所示,地理空间共分为 P 个等级,其中 L 级至 $M-1$ 级为全球尺度; M 级至 $N-1$ 级为区域尺度; N 级至 $P-1$ 级为城市尺度。将多种类型数据根据三个尺度进行重新组织,并分别选择 L 级、 M 级、 N 级存储该尺度下的元数据信息。

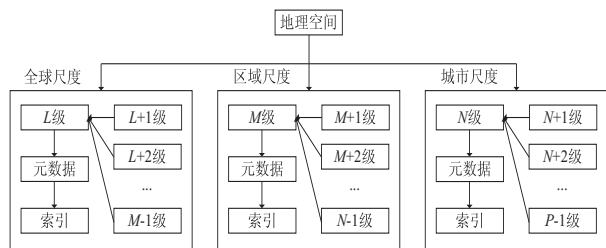


图 2 多尺度组织模型示意图

多尺度复合金字塔模型本质上是利用地理空间划分将同一尺度下不同类型元数据存储到统一的存储单元或单元组中,根据索引实现以空间区域为基础的文

件组织管理体系。采用这种数据组织与管理技术可以将用户访问数据的空间区域位置与数据文件本身表达的空间区域位置建立直接关联,提高了海量数据的寻址检索。

2 瓦片缓存机制

传统的缓存置换算法主要有:先进先出置换算法(FIFO)、最近最少使用瓦片置换算法(LRU)、最不经使用置换算法(LFU)^[10]。国内外学者针对瓦片数据缓存算法做了较多研究,Kang 等^[11]对瓦片预取和替换方式进行了研究,根据计算结果留下概率高的瓦片;在此基础上,张婷婷^[12]提出了基于时空热度的瓦片缓存置换策略,可以快速传输影像数据;Hsiao 等^[13]将多核 gpu 共享缓存中的瓦片进行分组,提出了一种基于多核位置感知的缓存置换算法(MLCR);史孝国^[14]通过考虑瓦片的访问频率及瓦片数据大小计算瓦片的保留价值,对置换算法进行了改进;涂振发等^[15]提出了一种最小空间数据价值缓存置换算法(GDLVF),基于时间、频率及空间位置计算瓦片价值,将价值最低的瓦片置换;刘佳星等^[16]提出了基于地理单元热度的瓦片缓存置换算法(GUH),通过考虑空间

特性,结合贪婪算法置换出热度值最低的瓦片;陆晔等^[17]研究了基于主题时空价值的瓦片缓存算法(GDTST),综合考虑了时间局部性、空间局部性以及用户主题倾向性,置换出主题时空价值最低的瓦片。传统的缓存置换算法没有考虑到瓦片数据的空间位置特性,瓦片命中率效果较差^[18];传统的缓存置换算法只考虑了瓦片进入缓存时间及命中次数,瓦片命中率较差。在上述研究中,依然存在如下三个问题:(1)缓存置换策略都是基于同一类型数据进行研究,不适用于多种类型数据进行加载;(2)部分研究人员虽然考虑了空间位置特性以及请求瓦片对下一时刻其邻近位置瓦片被访问造成的影响,但是在进行研究时没有考虑用户的操作习惯,请求瓦片对临近位置瓦片影响相同;(3)未设置保护机制,新进入缓存的瓦片由于价值低可能会很快被置换出去。

因此,该文提出了一种基于多尺度复合金字塔模型的缓存置换策略 MCPCR。MCPCR 满足多种类型数据在客户端进行缓存,并充分考虑瓦片数据的空间特性,根据用户操作习惯,动态计算当前命中瓦片相邻及相邻层级瓦片数据的请求频次,当瓦片数据进入缓存时,计算当前瓦片数据价值,当缓存空间已满或达到阈值时,缓存空间中价值最低的瓦片将被替换。同时引入瓦片保护机制,在瓦片进入到缓存空间后,一段时间内无法被置换,避免当缓存空间已满时,进行置换操作时将新进入到缓存空间的瓦片数据剔除。

2.1 瓦片缓存索引设计

当用户在客户端请求数据时,首先要在缓存中查找是否存在相关数据,因此,为了实现缓存中数据的快速检索,需要对缓存索引进行设计,从而提高数据检索速度,本次研究基于多尺度复合金字塔模型设计了索引机制。

服务器端为已经组织好的多分辨率层级的复合瓦片金字塔模型。当客户端请求相应数据时,当缩放层级为 Z 时,获取到当前显示区域中心点的地理坐标 $\text{Center}(X_c, Y_c)$;根据屏幕宽度 W_c ,屏幕高度 H_c ,以及浏览器显示的单位像素表示的实际地理距离 D_c ,以屏幕左下角为原点,计算屏幕左下角 (X_{LL}, Y_{LL}) 及屏幕右上角 (X_{UR}, Y_{UR}) 地理坐标:

$$\begin{cases} X_{LL} = X_c - \frac{W_c}{2} \times D_c \\ Y_{LL} = Y_c - \frac{H_c}{2} \times D_c \\ X_{UR} = X_c + \frac{W_c}{2} \times D_c \\ Y_{UR} = Y_c + \frac{H_c}{2} \times D_c \end{cases}$$

根据请求数据类型以及在该层级下地理范围对该类型瓦片数据行列号进行转换,并向复合金字塔模型中请求相应数据,将查询到的数据向客户端进行传递。

瓦片金字塔模型中瓦片的生成都是先将栅格数据或矢量数据进行切片,生成瓦片矩阵后再通过分级分块的方式构建多尺度瓦片矩阵集^[19],每张瓦片通过层级与行列号唯一确定。本次研究是对多尺度复合金字塔模型进行研究,具有多种类型瓦片数据,因此做出如下定义:

定义1:多尺度复合金字塔模型包含多种类型数据瓦片金字塔模型,每一张瓦片都可以通过数据类型、层级及行列号唯一确定。TileID 代表多尺度复合金字塔模型的瓦片索引值,可以表示为:

$$\text{TileID} = (\text{Type}, \text{Layer}, \text{Row}, \text{Column})$$

其中,Type 表示瓦片数据类型,Layer 表示该类型瓦片数据所在层级号,Row 表示瓦片在该级下的行号,Column 表示瓦片在该级下的列号。为了加快数据定位速度,索引项 TileID 采用哈希存储。基于多尺度复合金字塔的瓦片索引结构为:

$$\text{Index} = (\text{TileID}, \text{Value}, \text{Size}, \text{Frequency}, \text{LastTime}, \text{TimeInterval})$$

其中,Value 表示当前瓦片的价值,Size 表示瓦片所占空间的大小,Frequency 表示该瓦片的请求频次,LastTime 表示瓦片上一次被请求时间,TimeInterval 表示瓦片最近两次被请求的时间间隔。

2.2 MCPCR 策略

MCPCR 策略,当缓存空间已满或达到阈值时,置换出过了保护期限且价值最低的瓦片数据。其中瓦片价值为:

$$\text{Value}(a) = \frac{\text{Frequency}(a)}{\text{TimeInterval}(a)} \times \text{Type}(a)$$

其中,Frequency(a) 表示基于用户操作习惯的空间访问频次,TimeInterval(a) 表示当前瓦片的历史平均访问间隔,Type(a) 表示数据类型权重。

用户在客户端最常用的操作是平移操作和缩放操作^[20]。图3所示为瓦片相邻范围示意图。

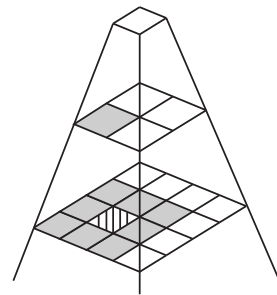


图3 瓦片相邻范围示意图

当用户操作为平移操作时,可以从上、下、左、右、

左上、左下、右上、右下八个方向请求瓦片数据;当用户进行缩放操作时,有放大操作和缩小操作两种情况。放大操作是指向服务器请求当前位置高层级数据;缩小操作是指向服务器请求当前位置低层级数据。也就是说,在某个时间用户访问了瓦片数据,附近的瓦片和相邻级别的瓦片数据在下次再次被访问的概率更高^[21]。故定义瓦片的请求频次 Frequency 为:

定义 2: 设 Frequency 为瓦片请求频次。根据用户操作历史库,分别记录用户操作 up、down、left、right、upperLeft、lowerLeft、upperRight、lowerRight、upLayer、nextLayer 及操作总次数 T 。当瓦片被命中时,此瓦片 $\text{Frequency} = \text{Frequency} + 1$,缓存中相邻瓦片数据及相邻层级瓦片数据 $\text{Frequency} = \text{Frequency} + t$,其中 t 为用户分别在 10 个方向上的操作占总操作次数 T 的比例。

Frequency 本质上是瓦片的累计访问频率及其相邻范围影响权值的总和。当瓦片被请求时,其相邻瓦片及相邻层级瓦片下一次被请求的概率增加,并根据用户操作习惯将不同方向上的瓦片赋予不同的权重。

通过计算瓦片历史平均间隔实现对瓦片数据命中在时间维度上产生的影响,同时考虑当前访问时间间隔以及历史访问时间间隔。故定义历史平均访问间隔 TimeInterval:

定义 3: 设 TimeInterval 为历史平均访问间隔,CurrentTime 表示当前瓦片访问时间,LastTime 表示瓦片上一次访问时间,TimeInterval 表示瓦片最近两次被获取的时间间隔, H_w 为历史访问权重, C_w 为当前访问权重。TimeInterval 公式为:

$$\text{TimeInterval} = [\text{TimeInterval} \times H_w + (\text{CurrentTime} - \text{LastTime}) \times C_w]$$

其中, H_w 与 C_w 的和为 1,当 H_w 的越大时,认为历史访问间隔对瓦片请求的影响越大。

由于数据类型多样,针对缓存置换策略的制定需要考虑到数据类型权重,同时考虑到瓦片单个数据大小,对数据类型权重进行定义:

$$\text{Type} = \frac{\text{Type}(i)}{\text{Type}(s)} * \frac{1}{\text{Size}}$$

其中, $\text{Type}(i)$ 表示访问的某一类型数据总和, $\text{Type}(s)$ 表示访问瓦片全部类型的总数量,并考虑瓦片大小对于缓存置换的影响,单个瓦片数据量越大,影响度越低,当缓存空间已满时,优先置换出数据较大的瓦片。

为了避免在缓存空间已满,新进入的瓦片由于价值最低,下一次置换操作时被移除,引入瓦片保护机制,也就是说,瓦片数据在刚刚进入缓存空间的一段时间内不能被替换。故定义瓦片保护时间 ProtectTime:

定义 4: 设瓦片保护时间为 ProtectTime,瓦片数据进入到缓存空间后初始化 ProtectTime,进入保护期,并随着时间的增加不断减小,当 ProtectTime 为 0 时,保护期结束,瓦片保护时间为 $\text{ProtectTime} = \text{ProtectTime} - \text{Time}$ 。

其中,Time 为当前时间与瓦片进入到缓存空间的时间差。

2.3 MCPCR 算法流程

(1) 新的瓦片数据进入缓存空间时,需要初始化瓦片信息,包括瓦片类型 Type,层级 Layer,行列号 Row、Column,价值 Value 以及保护时间 ProtectTime;

(2) 判断缓存空间能否容纳新瓦片,当缓存空间充足时,将瓦片存储在缓存空间中并执行步骤(3);当缓存空间不足时,则执行步骤(5);

(3) 将瓦片数据发送至客户端,同时得到该瓦片相邻瓦片数据信息及相邻层级瓦片数据信息,将已经在缓存空间中的瓦片数据执行步骤(4),否则一次性从服务器中获取所有瓦片;

(4) 更新缓存中瓦片价值 Value;

(5) 判断 ProtectTime 是否为 0,若已经为 0,说明已经过期,同时不再更新;若不为 0,则继续更新;

(6) 计算缓存空间中保护机制过期且价值最低的瓦片,按顺序进行移除直至缓存空间充足;

(7) MCP 算法结束。

3 实验与分析

3.1 实验环境

实验通过 Fiddler 采集客户端获取瓦片日志数据,根据获取到的瓦片数据计算用户操作,模拟用户瓦片数据操作记录,共计 1 267 154 条记录。其中最大的瓦片大小为 48.4 KB,最小的瓦片大小为 232 B。实验环境为 Intel(R) Core i5-8300H @ 2.30 GHz,4 核 CPU,12 GB 内存。根据用户请求记录,获得瓦片的类型、层级、行列号、大小及时间等数据,模拟用户在客户端的行为。

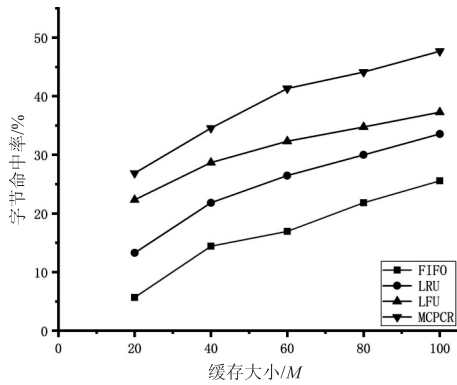
在 MCPCR 算法中,需要设置历史访问权重 H_w 以及当前访问权重 C_w ,通过实验测试,当历史访问权重设置为 0.7,当前访问权重设置为 0.3 时,实验效果较好。实验共由两个部分组成,首先分别选取三种数据类型单独进行实验,第二部分同时选取三种数据类型进行实验,统计缓存空间中的字节命中率及瓦片命中率,并传统缓存策略 FIFO、LRU、LFU 进行对比。

3.2 实验结果与分析

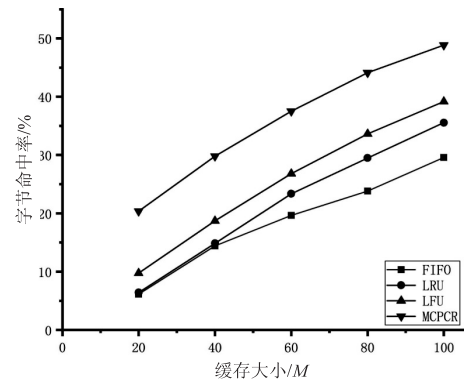
将用户请求的日志结果分别设为数据集 a、b、c、d,其中数据集 a、b、c 仅包含单一类型数据,数据集 d 包含 a、b、c 数据中三种类型数据。在设置客户端缓存

大小为 20 M、40 M、60 M、80 M、100 M 时,利用 FIFO、LRU、LFU、MCPCR 四种缓存置换策略进行模拟调度。比较相应的字节命中率及瓦片命中率,图 4 中(a)、

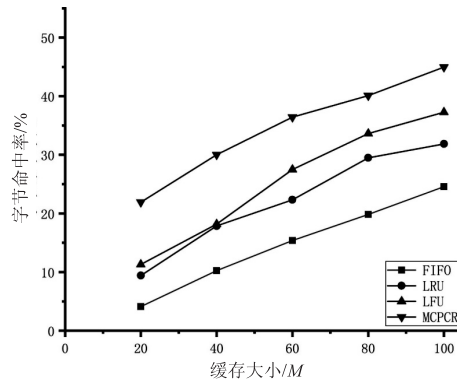
(b)、(c)、(d)分别代表数据集 a、b、c、d 字节命中率点线图,图 5 中(a)、(b)、(c)、(d)分别代表数据集 a、b、c、d 瓦片命中率点线图。



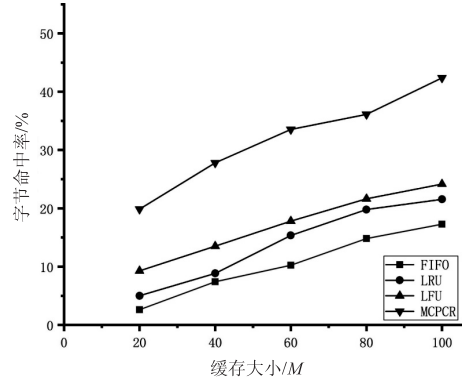
(a) 数据集 a 字节命中率



(b) 数据集 b 字节命中率

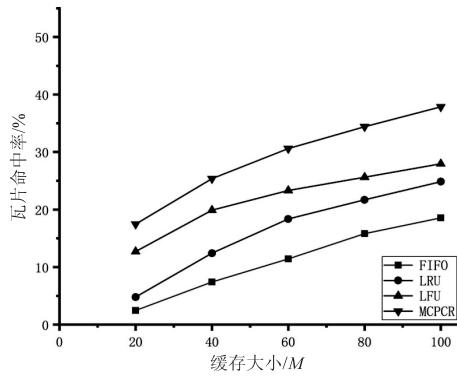


(c) 数据集 c 字节命中率

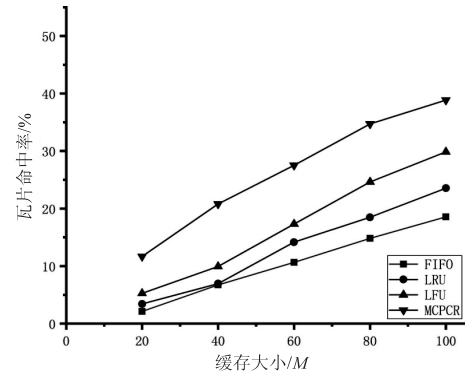


(d) 数据集 d 字节命中率

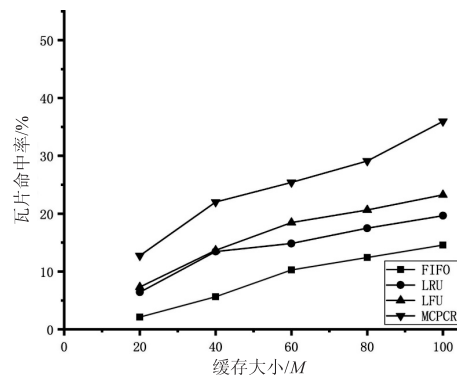
图 4 字节命中率点线图



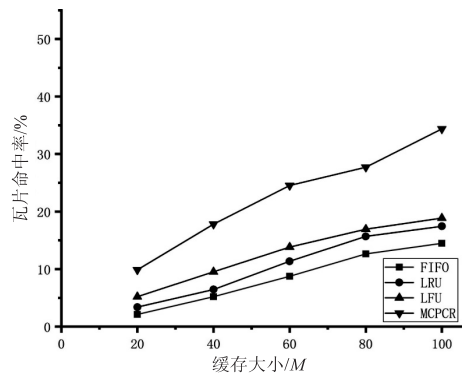
(a) 数据集 a 瓦片命中率



(b) 数据集 b 瓦片命中率



(c) 数据集 c 瓦片命中率



(d) 数据集 d 瓦片命中率

图 5 瓦片命中率点线图

通过实验可知,在三种不同的数据集下,四种缓存置换策略的命中率都会随着缓存空间大小的增加而逐渐增加,曲线最终趋近于平缓。由四种缓存置换策略进行对比可知,在对单一类型数据加载时,FIFO 缓存置换策略命中率最低,LRU 缓存策略淘汰最后被访问时间最久的数据,要优于 FIFO,LFU 缓存置换策略能够避免 LRU 周期性或者偶发性的操作导致缓存命中率下降的问题,整体优于 LRU,MCPCR 由于在考虑时间和空间因素的基础上,引入用户操作习惯因素及保护机制,在不同缓存空间大小下的命中率要优于其他三个缓存置换策略。在对不同类型数据进行加载时,传统缓存置换策略命中率下降明显,MCPCR 依旧能够保持良好的缓存命中率,通过实验证明,MCPCR 策略能够有效支持多种类型数据同时进行缓存,并能够有效提升缓存命中率。

4 结束语

该文提出了一种多尺度复合金字塔数据组织模型,能够有效地组织和管理不同类型的瓦片数据。基于该模型,设计了瓦片缓存索引。考虑到用户的操作习惯、历史访问对瓦片价值的影响并引入瓦片保护机制,提出了一种基于多尺度复合金字塔模型的缓存替换策略 MCPCR。实验结果表明,对同一种瓦片数据类型进行加载时,MCPCR 在不同大小的缓存空间中命中率均优于其他几种算法,在对多种类型瓦片数据进行加载时,MCPCR 仍然能够保持良好的命中率。可以有效支持不同类型瓦片数据加载,从而提高用户的响应速度。

参考文献:

- [1] 邵志伟. WebGIS 应用现状及发展趋势[J]. 化工管理, 2017(18):81.
- [2] 饶 鸣. 基于 WebGIS 的地理信息系统开发应用分析[J]. 信息系统工程, 2019(5):48.
- [3] 苏旭明,谭建成. WebGIS 中瓦片地图关键技术研究[J]. 北京测绘, 2012(2):9-12.
- [4] 谭庆全,毕建涛,池天河. 一种灵活高效的遥感影像金字塔构建算法[J]. 计算机系统应用, 2008(4):124-127.
- [5] 霍 亮,杨耀东,刘小勇,等. 瓦片金字塔模型技术的研究与实践[J]. 测绘科学, 2012, 37(6):144-146.
- [6] 王金地,霍 亮,慕 旭,等. 一种基于复合金字塔模型的三维模型数据组织方法[J]. 测绘与空间地理信息, 2019, 42(4):58-61.
- [7] 刘佳星. 基于矢量瓦片的 WebGIS 关键技术研究[D]. 北京:北京林业大学, 2017.
- [8] 褚 信,蔡阳军,杜震洪,等. 用户行为选择参与的五层十五级瓦片缓存置换策略研究[J]. 浙江大学学报:理学版, 2016, 43(4):452-457.
- [9] 曹梦鸽. 面向大数据的遥感影像金字塔模型的研究[D]. 哈尔滨:东北林业大学, 2018.
- [10] GENTA R, ERAND E, LGLI T. A comparison of three page replacement algorithms: FIFO, LRU and optimal[J]. Academic Journal of Interdisciplinary Studies, 2015(4):56.
- [11] KANG Y K, KIM K C, KIM Y S. Probability-based tile pre-fetching and cache replacement algorithms for web geographical information systems[C]//East European conference on advances in databases and information systems. Vilnius: Springer, 2001:127-140.
- [12] 张婷婷. 影像重采样 GPU 并行实现及瓦片缓存策略优化研究[D]. 杭州:浙江大学, 2018.
- [13] HSIAO C C, CHU S L, DAI S S. Efficient rendering and cache replacement mechanisms for hierarchical tiling in mobile GPUs[C]//2012 IEEE global high tech congress on e-electronics. Shenzhen, China: IEEE, 2012:170-175.
- [14] 史孝国. 基于一种改进缓存替换算法的校园电子地图研究[D]. 济南:山东大学, 2015.
- [15] 涂振发,孟令奎,张 文,等. 面向网络 GIS 的最小价值空间数据缓存替换算法研究[J]. 华中师范大学学报:自然科学版, 2012, 46(2):230-234.
- [16] 刘佳星,陈飞翔,陈星涵. 一种基于地理单元热度的瓦片缓存策略[J]. 计算机工程与应用, 2017, 53(5):90-96.
- [17] 陆 晔,张 伟,李 飞,等. 一种基于主题时空价值的服务器端瓦片缓存算法[J]. 浙江大学学报:理学版, 2020, 47(1):12-19.
- [18] LI R, WANG X, SHI X. A replacement strategy for a distributed caching system based on the spatiotemporal access pattern of geospatial data[J]. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2014, XL-4:133-137.
- [19] 路东林,智广玉. 地图发布平台下瓦片金字塔技术研究[J]. 数字技术与应用, 2013(3):99.
- [20] 李学俊,王尊亮,李晓欢,等. 基于地图切片预取的 WebGIS 系统[J]. 计算机应用研究, 2009, 26(10):3849-3851.
- [21] 苏焕焕. 海量遥感瓦片数据的快速发布服务研究与实现[D]. 郑州:河南大学, 2017.