

基于版本控制策略的 WebGIS 缓存实现机制研究

王映辉 冯德民

(陕西师范大学计算机科学学院 西安710062)

Study on Cache Implementation Mechanism of WebGIS Based on Edition Control Strategy

WANG Ying-Hui FENG De-Min

(Department of Computer Science, Shanxi Normal University, Xi'an 710062)

Abstract Analyze the access efficiency key factor of WebGIS, according to the locality and site-correlation, and by different access of client-side and server-side, combining mechanism of edition-control, advanced cache strategy and implementation mechanism of based client-side and server-side separately, and describe the detail of implementation based edition information. It can improve the efficiency of WebGIS greatly with this cache technology based on edition information.

Keywords WebGIS, Cache, Edition, Grid-slot

1 前言

基于 Web 的系统,有两个关键环节直接影响着数据的存取效率,即网络的传输效率和数据库的存取效率,Web 地理信息系统(WebGIS)也不例外。在 WebGIS 系统中,由于空间数据访问的局部性和位置相关性,鉴于性能考虑,使得构建在 Web 之上的 GIS 系统必须引入缓存技术。缓存技术已成为 WebGIS 实现中的研究重点之一^[1,2,6]。

当网络带宽和速度有限时,本地和网络数据访问在效率上存在着巨大的差异,决定了 WebGIS 系统与普通的 GIS 系统的不同,已成为地理空间数据在分布式环境中发布和共享的障碍。但是,利用对空间数据访问的局部性和位置相关性的特点,如果在本地建立数据缓存,暂时存储刚刚从数据服务器得到的空间数据及其相邻近的数据,则下一次访问这些空间数据时仅可以直接从本地缓存中读取,而不需要通过远程网络访问,则大大缩短了访问的时间,提高了访问的效率。

对 WebGIS 中数据库的存取效率问题,通过在服务器端设计专门的虚拟内存(内存文件缓存),并配合相应的缓存管理机制和规约来加速存取速度。

本文第2节分析了内存缓存、文件缓存和内存文件缓存三种缓存实现策略的优缺点;第3节按照图层、桶和实体数据组织和管理模型,给出了图层、桶和实体的版本标识和规约,进而较详尽地描述了基于客户端的缓存实现方法;第4节通过与客户端缓存实现技术的比较,简要阐述了基于服务器端的缓存实现技术;最后对全文进行了总结。

2 缓存的实现策略

内存缓存、文件缓存和内存文件缓存是缓存常见的三种实现策略。首先分析一下他们的优缺点。

2.1 内存缓存

内存缓存指在程序中开辟缓存空间,例如比较大的结构型数组等,将从数据库中得来的 GIS 数据(包括空间数据和属性数据)直接存放在内存中。其优点有:①对缓存容量随系

统虚拟内存的大小而变化,即缓存的容量受系统的限制小;②数据无需转换可以直接被访问和利用。内存缓存的局限性是:①虚拟内存的大量占用会降低系统性能;②同一程序的多次运行也无法共享缓存中的数据;③由于数据归具体的程序所有而受到了隔离,数据无法在多个同时运行的应用程序间共享。

2.2 文件缓存

文件缓存就是当内存缓存的容量超过一定限度时或是在应用程序结束时,把一些 GIS 数据存放到或存储到文件中,当再次访问这些数据时,再把它们装入内存。文件缓存的优点是:①同一应用程序的多次运行可以共享和利用缓存中的数据;②不占用大量虚拟内存,能够保证系统的性能。文件缓存的局限性是:多个同时运行的应用程序共享文件缓存中的数据时,每个应用程序的内存中都要有相同数据的副本,同时还要有很复杂的策略维护各个副本与文件缓存的一致性,进而增加了复杂度。

2.3 内存文件缓存

内存文件缓存是把内存访问和外存存储统一起来的一种缓存实现策略。具体地说,内存文件技术是指只需要对内存操作,而从文件中读取数据到内存和将内存数据写回文件的工作则由系统中的统一调度策略来完成。内存文件缓存的优点有:①当内存文件打开时可直接当内存访问,当内存文件被关闭时正好是磁盘上的文件。这样既不需要在文件系统和内存系统间进行转换数据,也不需要占用虚存资源,同时还可以长时间地保存缓存中的数据;②内存文件在不需要在多应用程序间进行数据副本的一致性维护的同时,便可完成数据的共享访问。内存文件缓存的缺点是:①增加了编程的复杂性。由于缓存中的数据存在文件中,两次打开相同的内存文件为其映射的基地址一般不同,各个数据的内存地址相应变了,因此只能用数组下标而不能用数据的指针描述数据间的关系,这会在一定程度上增加访问数据的时间和编程的复杂性;②容易形成读取垃圾或冗余。由于允许多个应用程序共享访问缓存的内存文件,所以必须避免共享访问带来的读取垃圾或冗

王映辉 博士,副教授,冯德民 教授,硕士生导师,主要研究方向为软件构架技术与算法理论。

余。

由于这三种策略都有各自的优缺点,在 WebGIS 中可综合使用它们。

3 WebGIS 客户端的数据库缓存的实现

3.1 缓存对象的标识与访问

在创建图层时,要为图层生成一个唯一的图层 ID 用来标识图层。当把图层数据读到缓存中时,根据图层 ID 分配不同的缓存空间。图层中的实体按照空间索引机制划分为若干个网格分区(桶)^[1],缓存从数据库读取图层数据时则以桶为单位进行。这是由于,一是对空间实体访问的位置相关性和局部性,绝大多数的查询是在一定的范围内进行的,而且也是位置相关的,这样,当图层数据量很大时,按照桶读取数据在满足查询要求的同时,可大大减少网络传输的数据量,降低网络负担。二是同一个桶中的实体数据由几何数据和属性数据两部分组成,由于不同的图层,实体几何数据部分的数据结构都是一样的,可以采用相同的管理模型;而实体属性数据却跟具体的应用密切相关,同时,这两部分数据的查询在很多情况下是分离的,加之几何数据比属性数据的稳定性高。因而分开缓存在满足查询要求的同时,可以提高缓存的效率。

缓存的实现大致包括缓存元素的选取、缓存存储和缓存访问三个方面。

在缓存实现时,缓存的基本单位是缓存元素,缓存元素在一般情况下包括桶的几何数据、桶的属性数据、图层信息、显示配置、图例库和参照系六类。对于每一类这种基本缓存元素都有唯一的标识码来标识,具体结构如表1所示。

表1 缓存元素的标识码

标识对象	标识码的组成
图层信息	数据库编号+“00”+图层号
图层显示配置	数据库编号+“01”+图层号
图层图例库	数据库编号+“10”+图层号
图层参照系	数据库编号+“11”+图层号
桶的属性数据	数据库编号+图层号+桶号+属性字段名
桶的几何数据	数据库编号+图层号+桶号

其中“+”号表示字符串连接运算。

在缓存存储的实现中,采用入口文件与数据文件相结合的缓存文件策略。入口文件的每一个记录中包括缓存对象的标识信息和该对象的版本信息;数据文件则可按照数据块队列链表的形式来组织维护,并反映数据块被访问的先后时间次序。

读和写缓存对象,是对于缓存访问的两个基本操作。读缓存对象通过对对象的标识,可利用散列函数等检索技术,定位对象在入口文件中的记录,如果入口记录不存在或者数据块记录已经被其它入口记录占用,读缓存对象失败。否则成功,并根据入口记录中的版本信息判断缓存数据的有效性,若有效,根据入口记录中的信息读取数据文件中的各个数据块,并进行组合,形成一个整体的数据;若版本陈旧,缓存数据无效,需从数据库中读取当前数据并更新缓存。写缓存对象的过程是分配入口记录、利用淘汰算法分配数据块、写数据、更新散列表等。

为了保证文件和内存的同步更新和缓存效率,对入口文件和数据文件可使用内存文件的形式。而对其它的缓存文件,为了保证系统的效率,则可采用内存缓存或文件缓存策略。

3.2 缓存更新中的版本及其规约

3.2.1 图层版本信息及其规约 在 WebGIS 的空间数据库中,图层表中的图层版本信息由5项组成:

①基本信息版本号(LayerVersion):记录图层的基本信息的更新程度,这些基本信息包括图层名、显示配置方案名、参照系名、图层分类信息和图层描述信息等;创建图层时置初值为1,每更新一次加1;

②入口版本号(EntryVersion):记录图层中实体号分配的更新程度。当实体被删除时,该实体的实体号成为无效实体号。为了保证无效实体号尽量地少,当无效实体号比较多时,对所有的实体号重新分配。创建图层时置初值为1,每重新分配一次实体号,EntryVersion 加1;

③空间索引版本号(IndexVersion):记录图层的空间索引参数的更新程度。创建图层时置初值为1,每重建一次空间索引,IndexVersion 加1;

④实体属性数据版本号(FieldVersion):记录图层中实体属性数据的更新程度。创建图层时置初值为0,每更新一次图层中一个或若干个实体的属性数据,FieldVersion 加1。增加实体或删除实体被认为是更新了实体的属性数据;

⑤实体几何数据版本号(GeometryVersion):记录图层中实体几何空间数据的更新程度。创建图层时置初值为1,每更新一次图层中一个或若干个实体的几何数据,GeometryVersion 加1。增加实体或删除实体被认为是更新了实体的几何数据。

不管是只有增加实体或删除实体或修改实体数据的操作,还是既有增加又有删除还有修改实体的操作,也不管一次增加多少实体或修改多个实体的数据,只要这些服务请求是一次提交给数据库的,就算作完成一次更新操作,这保证了某些情况下应用程序对于数据更新的事务性的要求。

3.2.2 桶版本信息和实体版本信息的变化及其关系与规约 桶版本信息记录在数据库里图层表的空间索引信息域中,是一个结构型数组(包括三个域:实体个数,实体属性数据版本号和实体几何版本号),顺序记录从桶0开始的所有桶中实体数据的更新情况,数组的下标就是桶号。

对桶中任何一个或多个实体的任何一次更新操作,对记录桶版本信息的结构型数组的修改规则如下:

①增加 n 个实体:实体个数加 n,实体属性版本号和实体几何版本号分别等于此次更新操作后图层的实体属性数据版本号和图层的实体几何数据版本号;

②删除 n 个实体:实体个数减 n,实体属性版本号和实体几何版本号的变化同上;

③只修改实体的属性数据:实体个数不变,实体属性版本号等于此次更新操作后图层的实体属性数据版本号,实体几何版本号不变;

④只修改实体的几何数据:需要按新的几何数据重新根据图层空间索引计算实体的桶号,若修改后实体的桶号等于原桶号,则实体个数不变,实体属性版本号不变,实体几何版本号等于此次更新操作后图层的实体几何数据版本号;否则,原桶的实体个数减1,新桶的实体个数加1,新旧桶的实体属性版本号和实体几何版本号都分别等于此次更新操作后图层的实体属性数据版本号和实体几何数据版本号;

⑤修改实体的所有数据,则进行上面变化情况的并操作。

3.3 缓存有效性的版本控制算法描述

缓存中对图层的实体数据按桶存取的数据读取策略,使

得缓存中的图层数据在很多时候只有部分有效性。对图层而言,某一时刻缓存中可能只有几个桶的实体数据是有效的,但同一时刻同一个桶内的实体数据的有效性一定是相同的,即同一个桶内,所有的实体数据要么都有效,要么都无效。

这样,判断缓存中图层数据的有效性可以分为三个层次来进行,各个层次数据的有效性取决于不同的图层版本信息:①图层基本信息,即图层基本信息版本号;②图层的空间索引信息,即图层空间索引版本号;③图层实体信息,即图层入口版本号、图层实体版本号(几何数据与属性数据版本号)以及各桶的版本号。判断的步骤及更新策略如下:

第一步,如果缓存中图层的基本信息版本号不等于数据库中图层的基本信息版本号,说明图层的基本信息如图层名、显示配置等被重新分配过了,需要更新缓存中该图层的基本数据,否则不用更新缓存,则:

第二步,如果缓存中图层的空间索引版本号不等于数据库中图层的空间索引版本号,说明图层的空间索引已被重建,可能所有实体的桶号都发生了变化,此时缓存中该图层的所有实体数据全部无效,需要更新该图层的空间索引数据和所有的桶及其实体数据;如果两个空间索引版本号值相等,则:

第三步,如果缓存中图层的入口版本号不等于数据库中图层的入口版本号,说明图层中所有实体的实体号被重新分配过了,需要更新缓存中该图层的所有桶及其实体数据;否则:

第四步,如果缓存中的实体属性数据版本号 FeatureDataV1 不等于数据库中该图层的实体属性数据版本号 FeatureDataV2,或者缓存中的实体几何数据版本号 GeometryDataV1 不等于数据库中该图层的实体几何数据版本号 GeometryDataV2,说明图层中有某些桶的实体数据被修改过了,需要将缓存中该图层所有桶中满足下面条件:数据库中该桶的实体属性数据版本号在 FeatureDataV1 和 FeatureDataV2 之间、或者实体几何数据版本号在 GeometryDataV1 和 GeometryDataV2 之间的所有实体数据进行全部更新。

另外,缓存分为内存缓存和外存缓存。内存缓存是动态临时的,当程序运行时,访问同一个图层的多个进程可以共享该图层的内存缓存,当程序结束时,需要把内存缓存中的数据交换到外存永久缓存中;在程序重新启动时,又可以把外存缓存交换到内存缓存中。

不管是内存缓存还是外存缓存,容量都是有限的。当缓存空间消耗完时,一般根据数据访问的频度和未访问时间来淘汰缓存,即采用最近最少被访问 LRU (Least Recently Used) 的数据最先被淘汰策略。

4 WebGIS 服务器端的数据库缓存的实现

WebGIS 系统中,所有的客户端通过 WebGIS 服务器从

WebGIS 空间数据库中读取和写入数据,而多个客户端向服务器的并发访问,其对数据请求的重合度极高甚至完全相同,如果能重用以前刚刚访问过的数据库数据,就免去了与数据库的交互,大大提高数据访问的效率,这可采用 WebGIS 服务器端的缓存技术来实现。

在服务器上实现缓存与客户端是不尽相同的。大概包括以下几个方面,首先是在结构上,客户端缓存的最高一级的单位划分是图层;服务器上由于要管理多个数据库连接,其缓存的最高划分应该是基于不同的数据源;其次是在缓存读取上,客户端上的缓存在读数据时可以通过比较版本以判断缓存数据的有效性,即在客户端缓存中,当缓存数据无效时,才转向远程服务器索取数据,然后再更新缓存。而在服务器上,可以采用客户端对服务器缓存里读和写数据的双向缓存技术。

与客户端缓存技术不同的是,由于只有一个 WebGIS 服务器,而且服务器缓存是数据进出数据库的唯一通道,要实现高度一致的服务器缓存,可以采用主动更新的方法,即不管客户端是否有服务请求,服务器根据缓存的内容定时从数据库读取最新数据,以便及时保证缓存的内容和数据库的一致性。

结束语 本文针对存取效率问题,依照 WebGIS 对空间数据访问的局部性和位置相关性,给出了 WebGIS 的缓存实现策略。首先分析了影响 WebGIS 的存取效率的关键因素;随后给出了基于图层、桶和实体的数据组织模型,通过桶将空间图层与实体进行了关联,增加了对空间实体管理和检索的灵活性,同时,描述了图层、桶和实体的版本编码和相互间的关联规约;最后,针对客户端和服务端对数据存取方式的不同,结合版本机制,分别给出了基于客户端和基于服务器端的版本控制缓存实现策略,并对其细节进行了描述。实践表明,这种缓存技术可大大提高 GIS 在 Web 上对空间数据的存取效率。

参考文献

- 1 Xiao Weiqi, Feng Yucui. Grid Spatial Index Mechanism for Spatial Object Database. Chinese Journal of Computers (in Chinese), 1994, 17(10): 736~742
- 2 吴信才,等. 地理信息系统原理与方法[M]. 北京:电子工业出版社,2002
- 3 汤国安,赵牡丹. 地理信息系统[M]. 北京:科学出版社,2000
- 4 施伯乐,等. 数据库系统教程[M]. 北京:高等教育出版社,1990
- 5 承继成,等. 国家空间信息基础设施与数字地球. 北京:清华大学出版社,1999
- 6 王映辉,周明全,耿国华. 基于 GIS 的“数字城市”分布式协作模型研究. 计算机工程与应用,2002,38(8): 20~22
- 7 王映辉. 一种 GIS 自适应层次网格空间索引算法. 计算机工程与应用,2003,39(3): 58~60