文章编号:1671-4598(2016)05-0209-04 DOI:10.16526/j.cnki.11—4762/tp.2016.05.059 中图分类号:TP311 文献标识码:A

WebGIS 中带图业务数据的缓存和预取机制研究

李 源,何友全
(重庆交通大学 信息科学与工程学院,重庆 400074)

摘要:目前在 WebGIS 中地图缓存已得到深入的关注和研究,随着 WebGIS 客户端性能的逐渐增强,如图片音频等越来越丰富的内容也能被展示出来;由于服务端磁盘读取和网络传输两大瓶颈,一定程度上延长了用户访问这些资源的等待时间;以某高速公路管理系统为背景系统,采用了在服务端和客户端各加一组业务数据缓存、客户端启动后预先读取服务端缓存的方式,提高了带图片的业务数据的访问性能;同时根据该 WebGIS 中业务数据特有的地理位置关系,在分析用户访问轨迹后,客户端预测用户可能访问的下一条数据并预取,进一步提高了缓存性能。

关键词:网络地理信息系统;缓存;预取

Research on Caching and Prefetching Business Data with Pictures in WebGIS

Li Yuan, He Youquan
(School of Information Science&Engineering, Chongqing Jiaotong University, Chongqing 400074, China)

Abstract: Currently in WebGIS map cache has been deeply concerned and researched. With the increase of WebGIS client performance, WebGIS can show more rich content, such as pictures, audio and other media. The server reading disk and network transmission are two major bottlenecks, have partly extended the delay of user access to these resources. A highway management system for the background of the system. Added sets of business data caching in server and clients, and enable clients prefetch server's cache after started up, have improved the performance of access business data with pictures. According the geographical relations of business data in thisWebGIS, after analyzed users' accessing tracks, prefetching the next data that users may access improved cache performance further.

Keywords: WebGIS; cache; prefetch

0 引言

在 Web 应用中,加入缓存(Cache)的目的是降低数据访问延时,节省网络带宽。缓存的作用是将常用的或者感兴趣的信息暂存本地方便调用,而不必从远程的数据源获取。缓存的内容是过去访问的历史数据,根据过去的访问规则决定缓存队列的更新与否。与 CPU 缓存不同的是,Web 缓存可以部署在客户端、服务端或者是介于客户端与服务端之间的网关和代理上面。目前缓存在 WebGIS 的应用主要是地图数据的访问方面[1-3]。相对地图切片这类约数十 KB 的数据量而言,当业务数据中加入了图片、视频等数百 KB 甚至数 MB 的内容时,这些内容将占据读取和传输的大部分时间,因此对这些业务数据进行访问优化是非常必要的。

与缓存这类被动暂存数据的形式不同,预取(Prefetch)利用的是网络、客户端的空闲时间,即客户端与服务端常规性交互访问结束后,等待用户响应的时段进行预先主动缓存。从而既降低了用户一般性访问的延时,又不会额外增加网络负担。

本文就某公路管理 WebGIS 中带图片业务数据内容(如图 1),在客户端与服务端两端加入缓存与预读取机制以期在理想情况下应去除所有的等待时间,尽可能地减少用户在访问远程数据的过程中感知到的延迟,提高访问效率。根据公路管理系统中业务数据在空间上存在线型关系的特点(如图 2),采集 30 名用户的业务数据点的访问轨迹,随机平均分为两组:一组用于设计预读取机制;另一组用于检验该机制的效率。

图 1 需要被展示的图片及数据示例

图 2 各数据点空间线型关系示例

1 双缓存的建立

考虑整个系统数据访问瓶颈主要在两个方面:1)服务端将数据从磁盘上的数据库中读取到内存;2)客户端通过网络从服务端获取数据。因此,要充分降低系统访问数据的代价,则需要同时对这两个操作进行优化。本文所采用的方法是在分别在服务端与客户端中建立各自的缓存,其中,服务端缓存根据所接收到的全部请求建立,客户端缓存根据用户发出的查询请求建立。

收稿日期:2015-11-10; **修回日期**:2015-12-15。

作者简介:李 源(1989-),男,湖南省津市人,硕士研究生,主要从事 WebGIS 开发方向的研究。

何友全(1964-),男,湖北荆州人,教授,硕士研究生导师,主要从事数据挖掘、软件工程方向的研究。

1.1 客户端缓存的建立

对于客户端缓存的大小，考虑在当前环境下，无论是传统的计算机还是智能手机，所配备的内存容量已经比较充足。以某智能手机系统为例，一个浏览器进程可以获得约 300 MB 的运行内存空间，本文中图片数据的平均大小约为 500 k 字节，综合缓存队列维护等其他因素，本文设客户端缓存队列大小为 96 个，即约 50 MB 的内存空间大小。

为减轻客户端的运算压力，缓存替换算法选用易于实现、复杂度低且命中率较高的 LRU（最近最少使用）算法。

1.2 服务端缓存的建立

考虑在使用本系统过程中存在多用户集中访问如新增数据、异常数据等局限于某几个点的数据情况，鉴于有“热点”数据的存在，同时为后文所提出的预取作基础，本文在服务端建立两级缓存：第一级用于缓存“热点”数据，队列大小为 32 个；第二级用于缓存最近访问过的 512 条历史数据。服务端缓存算法流程如下：

1）数据第一次被访问，投入第二级缓存队列内；

2）若该数据在第二级缓存队列内没有被访问 3 次，则按队列淘汰规则 FIFO（先进先出）淘汰；

3）若该数据在第二级缓存队列内访问次数达到 3 次后，移入第一级缓存；

4）数据在第一级缓存内，按 LRU 规则进行排序与淘汰，需要被淘汰时，不移入第二级缓存。

为进一步提高一级缓存所缓存的“热点”数据的准确性，排除由于单用户频繁操作导致的伪“热点”。本文在第二级缓存中的每个数据附加了两个属性字段记录最近访问该条数据的用户标识，若用户标识未被记录，则记录，否则相同标识的用户的访问不被记录。当数据第三次被不同标识的用户访问时，则移入第一级缓存队列内。

1.3 一致性维护

为了保证数据更新后，所有缓存与数据库中存储的数据保持一致，需要在数据更新操作完成时，由数据库通知缓存进行数据同步。本文采用的通知方法是数据库内有任何数据改动后，激活触发器（Trigger）将所更改的数据复制到 ChangeNotificationCacheTables 表内记录已经更新的数据，服务端抽取 ChangeNotificationCacheTables 表的内容，根据该表内的内容进行缓存的更新操作，若更新的数据存在于缓存内，则更新，否则忽略。

由于客户端相对于服务端而言处于远程，不能同服务端一样可以方便地进行内容检测和更新。对于客户端的缓存的一致性维护主要有两种方法：一是由服务端通知客户端，即服务端维护一张客户端列表，当有数据更新时由服务端向所有客户端进行广播式通知，缺点是客户端列表不易维护，广播式通知资源消耗较大；二是客户端询问服务端，客户端通过定时或固定情况下向服务端查询自身缓存内数据是否需要更新。若客户端频繁检测缓存内容的更新与否将耗费大量时间与带宽。本文以节省不必要的网络消耗和提高用户体验两大主要目标，本文采用先显示，再更新的机制：当用户访问的信息在缓存中且恰好数据库中的对应数据被更新时，先将未更新的数据显示出来，减少呈现给用户“加载”时间，同时向服务端发出更新请求，服务端检测到数据不一致，返回给客户端更新后的数据，最终客户端更新缓存并刷新客户端正在显示的内容。本文中用于标识数据特征的方法采用 CRC16 校验码的方法，将一个点的所有数据转换为字节流，计算该字节流的 CRC16 值。服务端根据接收到请求中的主键和 CRC16 值计算数据库中对应的数据的 CRC16 值，若一致，向客户端答复一致；否则答复不一致并附带更新后的数据。

2 客户端预取

为在客户端进一步提高命中率，特别是针对客户端刚开始运行时缓存为空的情况。本文在客户端方面使用了两种预取方法，一种是在客户端完成启动时立即从服务端获取服务端的一部分缓存；另一种是预测用户的访问顺序，预测用户将要访问的下一个资源，提前从服务端获取并缓存。客户端预取的数据同用户普通访问的数据一样投入缓存队列并进行队列更新。

2.1 启动时预取

如图 3 所示，传统的客户端工作模式是被动的，程序启动完毕后，总是在等待用户操作，直到用户完成第一次操作发出操作请求，客户端才与服务端交互返回用户响应的结果。而与用户操作相比，程序响应的速度通常更快，因此，被动等待用户操作将浪费大量时间，同时程序的响应时间也得不到改善。

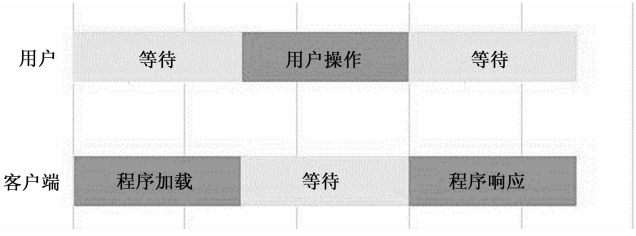


图 3 传统工作模式示意甘特图

通过分析用户使用习惯得出，客户端程序在浏览器内加载完毕后，总是经历了（用户反应程序就绪——输入认证进行身份认证——认证成功——发出第一次查询请求）这样一个过程。经统计，上述过程平均时长约为 13 秒。考虑身份认证步骤消耗资源极少，本文利用客户端加载完毕到用户第一次查询请求发出的短暂时间进行数据预取。预取的内容，本文所采取的方法是预取服务端缓存中的第一级缓存中的全部内容，即获取其他用户所访问过的最频繁的数据作为客户端的初始缓存，有利于提高客户端缓存的初始命中率。

2.2 预测访问顺序预取

在 WebGIS 中，对于各业务数据的相关性，主要分为两类，一类与传统管理系统类似的根据某一数据属性排列的关联如：ID、上下行方向、采集日期、记录人员等；另一类则不同于传统管理系统的业务数据，地图上的业务数据点与点之间存在空间上的相邻的关系。在本文所涉及的系统中，各点在空间上的排列是沿高速公路的走向排列的，即：除了最两端的点外的所有点仅在某一方向上存在一个前驱点，在某一方向上存在一个后继点。

通过所采集的若干人的访问轨迹统计后得出：访问地图某一点后，访问相邻点的概率为 56.4%；点间平均访问间隔 2.86 秒。考虑网络传输因素，2.8 秒内可传输约 3 个业务数据

点的全部数据。因此，若在用户访问某点的同时，客户端后台提前从服务端读取当前点的两个相邻点数据，则在理论上，下次访问的缓存命中率将达到 56.4%。特别是当用户是沿某一个方向访问数据时，缓存命中率将更高。

综上，客户端的工作模式为图 4 所示。

图 4　预取机制下工作模式示意甘特图

3　测试与分析

对证明缓存对访问性能的提升，本文采用重复访问轨迹法，随机排列测试组访问轨迹组成测试输入，连续访问测试输入，分别对不使用缓存、仅使用客户端缓存、仅使用服务端缓存和使用双缓存这 4 种情况进行测试，记录每次访问所需要的时间、内存占用以及 CPU 占用。测试软硬件环境为：

服务端：CPU：IntelCorei5-3210 M，内存大小：8 GB，操作系统为 Windows10 专业版 64 位，数据库管理系统为 SQL server 2008R2，GIS 服务使用 ArcGISServer10.2；客户端相同配置 2 台：CPU：IntelCorei5-4300U，内存大小：8 GB，操作系统 Windows10 专业版 64 位，浏览器为 InternetExplorer11 并启用 Silverlight 插件；每个客户端中额外使用 Hyper-V 技术运行一台虚拟机，分配 50%的 CPU 时间及 4 GB 内存，运行 Windows8.1 专业版 64 位；网络环境为百兆局域网。

为保证测试准确性，模拟多用户同时使用的环境，测试时，4 个客户端同时启用进行访问测试，并输出每次访问的时间戳，使用 VisualStudio 的调试功能，记录各客户端的内存和 CPU 占用。最终将 4 个客户端的全部记录统计汇总后如下表所示。

表 1　4 种情况下的访问代价

	平均访问花费时间	最大访问花费时间	平均内存占用大小	平均 CPU 占用率
不使用缓存	267.3 ms	283.8 ms	61 MB	7.1%
仅客户端缓存	176.4 ms	284.1 ms	135 MB	9.2%
仅服务端缓存	215.6 ms	260.1 ms	60 MB	7.9%
使用双缓存	134.8 ms	301.3 ms	132 MB	8.6%

由表中数据可知，不使用缓存时，每次访问都将执行一次完整的请求－提交－读盘－传输返回过程因此平均访问时间和最大访问时间相差不多；仅使用客户端缓存时，由于服务端数据库磁盘读取未被优化，因此当客户端缓存访问命中时，访问时间才被缩短；同理，仅使用服务端缓存，则客户端每次访问的数据都需要通过网络传输，因此仍然存在相对较长的延时。仅当使用双缓存时，两端均得到了优化，对访问时间有较明显的缩短。

本文在原系统中进行上文所述的将本文所采集到的另一组用户访问轨迹用户进行对应测试从命中率、响应速度和网络占用三方面同传统 LRU 缓存算法进行对比，证明本文所述相关方法的有效性。

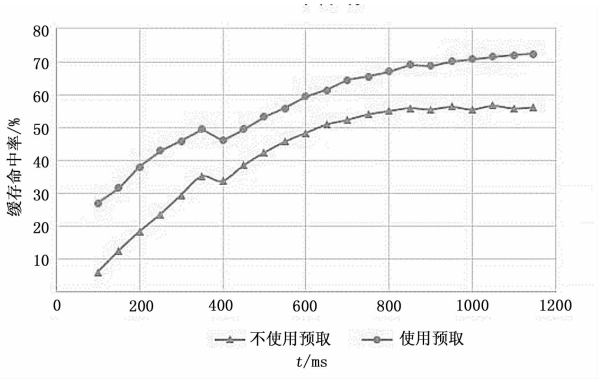


图 5　使用预取前后客户端缓存命中率曲线

从图 5 中曲线可以看出，在不使用预取的情况下，客户端缓存命中率从 0 开始上升，经过较长时间的使用后，缓存命中率维持在 55%左右；而在客户端加入预取机制后，由于初始预取数据的存在，缓存初始命中率就为 25%，是不使用预取情况下访问 250 次达到的效果，经过较长时间访问后，缓存命中率在 70%以上。说明由于预取机制的加入，缓存命中率有了较大的提高。

表 2　预取和不预取下的访问延时及网络占用

	平均访问时长	最大访问时长	平均网络速率	最大网络速率
预取	105.7 ms	326.8 ms	1.32 MB/s	1.57 MB/s
不预取	136.0 ms	275.6 ms	773 kB/s	1.29 MB/s

由表中数据可得，虽然加入了预取机制导致最大访问时间有所增加，但由于缓存命中率的提高，平均访问时间得到减少；网络方面，由于预取机制在用户反应及操作时间段客户端后台仍然在向服务端请求业务数据，网络间歇时间比不预取短，因此网络使用率高于不预取的情况。

同时，对于 1.57 MB/s 的峰值网络速率，使用约 12 Mbps 带宽的网络即可满足传输需要。当前蜂窝移动数据网络 3 G 标准中的 HSDPA 技术提供 14.4 Mbps 的带宽，可满足本系统的移动使用要求，符合预期的优化效果。

4　结束语

本文基于传统 WebGIS 业务系统，针对传输单元大的带图业务数据在客户端和服务端两方面进行了访问优化，为两端设计了不同的缓存方法，通过重现用户访问轨迹的方式进行了测试，取得了较好的效果，主要优点如下：

（1）服务端采用两级缓存，尽可能地保证了所有用户中访问最频繁的数据保留在缓存队列内；

（2）另外在客户端加入了主动预取机制，提高了客户端初始缓存命中率和网络利用率，降低了访问延时。

参考文献：

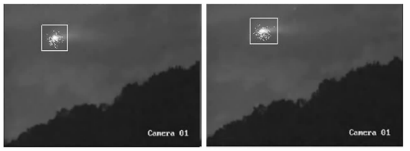
[1] 魏祖宽，胡娟，金在弘. WEBGIS 混合缓存的应用与研究 [J]. 计

算机系统应用，2009 (9)：144－148.
[2] 祁 羽，陈 荦，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，2007，29 (5)：41－44.
[3] 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.
[4] Deng Y F，Manoharan S. A Review of Web Cache Prefetching [J]. Journal of information and communication convergence engineering，2014，12 (3)：161－167.
[5] Yeşilmurat S，işler V. Retrospective adaptive prefetching for interactive Web GIS applications [J]. GeoInformatica，2012，16 (3)：435－466.
[6] Li R，Guo R，Xu Z，et al. A prefetching model based on access popularity for geospatial data in a cluster-based caching system [J]. International Journal of Geographical Information Science，2012，26 (10)：1831－1844.
[7] Johnson T，Seeling P. Web cache object forwarding from desktop to mobile for energy consumption optimizations [A] . Green Communications (OnlineGreencomm)，2014 IEEE Online Conference on [C] . IEEE，2014：1－7.
[8] 徐 超，曾学文，郭志川. 基于资源预测的智能终端资源缓存算法 [J]. 计算机工程，2015，41 (3)：59－63.

(上接第 208 页)

3 仿真结果分析

本实验在具有英特尔 I5 处理器，内存 4 GB 的笔记本进行，数据分析采用 MATLAB 7 进行。红外摄像机镜头焦距为 25 mm，拍摄距离大于 80 m，环境照度为 0.05～0，1 Lux。在试验刚开始时人为给出初始点，如果锁定目标了，对目标采用白色框进行标示。



(a) 第2帧 (b)第26帧
(c)第57帧 (d)第93帧

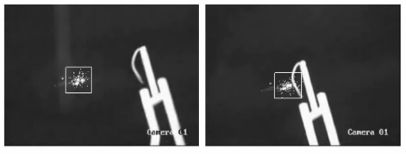
图 2 第一组测试序列效果分析图

试验中我们在红外的情况下对飞行模型进行跟踪，每帧图像像素为 290×210，红外目标灰度分布等级为 8，$N=100$。第一组实验在无遮挡条件下，对红外飞机目标的进行追踪，共测试 100 帧，在粒子滤波算法对飞机状态进行粗略估计之后，用均值漂移算法进行进一步精确搜索。从图 2 中可看出，本文设计的算法可以精确稳定的进行红外目标跟踪。

第二组实验，在遮挡情况下进行红外飞机目标的追踪，试验测试共 50 帧。图 3 (c) 第 25 帧时，飞机明显被遮挡，本文算法融合 Mean shift 的抗遮挡特性，在有遮挡的情况下，再结合 Mean shift 算法模型，在进一步缩小的区域内执行再查询，试验中在 32 帧后精确的追踪上目标。结合试验结果，对比其他的设计本文的设计具较强的鲁棒性，还减少了算法计算量，提高了红外目标跟踪的实性。

4 结束语

基于统计类框架下的粒子滤波与基于优化类框架下的均值漂移，各自都难以表现出良好的跟踪性能，本设计根据两类跟踪框架的优点，设计了一种在粒子的滤波模型下增加均值漂移模型的跟踪算法。同时，针对红外系统探测距离远，目标成像小而导致的目标特征不明显、目标背景复杂、遮挡等问题，提取红外图像的灰度信息，根据巴氏系数自适应更新模板，使红外目标在遇到遮挡时仍能鲁棒地跟踪。若想要更好得实现复杂环境下红外目标的追踪鲁棒性，可融合多个红外信息特征对目标进行追踪。



(a)第8帧 (b)第19帧
(c)第25帧 (d)第32帧

图 3 第二组测试序列效果分析图

参考文献：
[1] 姜锦锋. 红外图像的目标检测、识别与跟踪技术研究 [D]. 西北工业大学，2007.
[2] 姬雪峰. 基于粒子滤波的红外目标跟踪方法研究 [D]. 西安：西安电子科技大学，2011.
[3] Comaniciu D，Ramesh V，Meer P. Kernel－based object tracking [J]. IEEE Trans on Pattern Analysis and Machine Intelligence，2003，25 (5)：564－575.
[4] Doucet A，Gordon N，Krishnamurthy V. Particle Filters for State Estimation of Jump Markov Linear Systems [J]. IEEE Trans. Signal Processing，2001，49 (3)：613－624.
[5] 管小清，吕志强. 适用于复杂环境下的实时目标跟踪技术 [J]. 计算机测量与控制，2012，20 (10)：2776－2778.

2016 年 1 期文章《基于车载电子标签数据的单交叉口状态判别研究》，增加第四作者，作者信息为：
姓名：刘续博
所属单位：美国密歇根大学 工业与运营工程学院
College of Engineering University of Michigan