

基于多线程与缓冲池的 WebGIS 数据传输

王江鹏¹, 李先国²

(1. 西北工业大学软件与微电子学院, 西安 710072; 2. 西北工业大学计算机学院, 西安 710072)

摘要: 图像数据传输速率较低是 WebGIS 发展的主要障碍之一。针对地图分层、图像分割中的图像数据, 采用多线程技术提高系统吞吐量、有效利用系统资源并管理多用户请求, 通过缓冲池技术提高服务器主机性能, 减少传输时的磁盘搜索次数。基于 C#.net 实现对 WebGIS 图像数据传输的优化, 提高了数据传输效率。

关键词: 多线程; 缓冲池; 数据传输

Data Transmission of WebGIS Based on Multithread and Buffer Pool

WANG Jiang-peng¹, LI Xian-guo²

(1. College of Software and Microelectronics, Northwestern Polytechnical University, Xi'an 710072;

2. School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710072)

【Abstract】 Low efficiency of image data transmission is one of the main drawbacks during WebGIS developing process. Aiming at image data of map delamination and image division, this paper uses multithread technology to increase the system throughput, use the system resource effectively and manage multiuser request. It improves the host performance and decrease the disk search times by buffer-pool technology. Based on C#.net, image data transmission of WebGIS is optimized and the data transmission efficiency is improved.

【Key words】 multithread; buffer pool; data transmission

1 概述

在地理信息系统(Geographical Information System, GIS)中, 现实世界被表达成一系列地理要素和地理现象。近年来, 随着国际互联网的发展, 结合了网络技术和 GIS 技术的网络 GIS 成为地理信息系统的重要发展方向之一。

地理数据的网上传输是 GIS 网上发布的关键, 如何实现以地图为载体的地理信息网上传播是 WebGIS 要解决的一个核心问题。本文涉及的地理数据以地图分层和图像分割的方式进行组织, 在 C#.net 下使用多线程及缓冲池技术实现 WebGIS 的数据传输, 提高了空间数据的传输速度。

2 WebGIS 数据传输

2.1 WebGIS 结构模型

目前应用较普遍的 WebGIS 结构模型采用由数据库服务器、GIS 服务器和客户端组成的 3 层 C/S 或 B/S 结构, 见图 1。

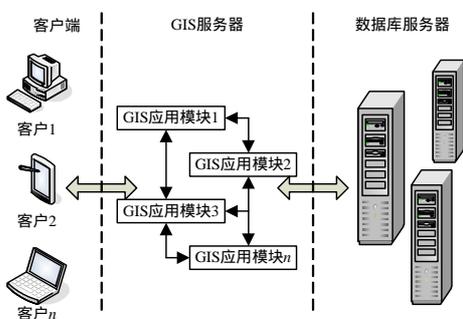


图 1 3 层 WebGIS 结构模型

2.2 WebGIS 数据传输方案

目前 WebGIS 的数据传输^[1]主要采取 2 种方式:

(1) 以 tif, jpeg 等图像格式传输。用户在图像上移动鼠标

时, 程序得到鼠标的屏幕坐标位置, 并将其返回给服务器处理。服务器先根据屏幕坐标计算出地图上的实际位置并确定用户的操作或选中的空间特征或对象, 然后完成用户指定的操作, 最后将结果以图像形式返回到客户端。

(2) 以矢量图形的方式传输。客户端得到的是矢量数据, 而不是一幅图像。用户在图形上移动鼠标时, 在客户端上即可实时判断出鼠标所指的是哪一个点、线或多边形, 能改变其颜色或以高亮度显示, 并将代码或名称传给服务器。

本文以地图分层、图像分割的图像数据为研究对象, 如图 2 所示, 在地图数据传输流程中, 左端为客户端, 右端为数据库服务器, 位于中间的 GIS 服务器是本文讨论的重点, 包括一个为处理每个客户端请求所创建的线程池, 以及用于提高搜索传输效率的数据缓冲池。

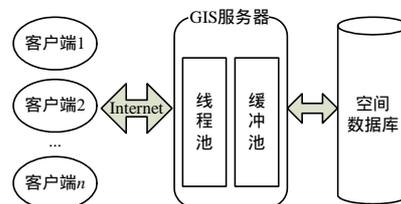


图 2 WebGIS 数据传输流程

3 基于 C#.net 多线程的图像数据传输

3.1 C#.net 多线程技术

多线程指允许单个程序创建多个并行执行的线程来完成各自的任务。在多线程程序中, 当一个线程必须等待时,

作者简介: 王江鹏(1984 -), 男, 硕士研究生, 主研方向: 软件工程; 李先国, 副教授

收稿日期: 2009-08-12 **E-mail:** jiangpengwang@hotmail.com

CPU 可以运行其他线程而不用等待，因此，大大提高了程序效率。

在 .net 类库^[2]中，与多线程机制应用相关的类都放在 System.Threading 命名的空间中。其中，Thread 类用于创建线程；ThreadPool 类用于管理线程池。该类库提供了解决线程执行安排、死锁、线程间通信等实际问题的机制。

C# 的特点使用户可以轻松使用它对多线程进行控制，简化了编写代码的工作。

3.2 服务器端线程池设计

在常见的 WebGIS 系统中，针对大量用户的并发访问，为确保系统响应速度和对服务器资源的最少占用，通常采用线程池技术^[3]，在服务器端为每个用户创建一个相应的线程，有效管理多用户的并发访问。图 3 给出了多用户并发访问多线程机制。

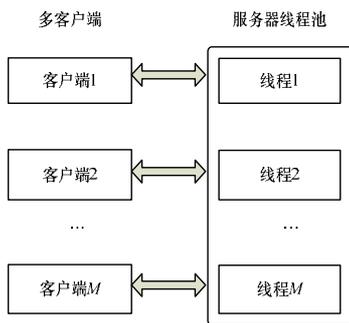


图 3 多用户并发访问多线程机制

本文在上述机制的基础上，设计了一种改进的多线程传输方案：在单用户线程下，为某一时刻用户向服务器发送的每个数据请求，创建相应的数据传输子线程。

用户数据请求如下：假设某一时刻客户端屏幕显示 M 幅图像，该客户端向服务器发送对应的 M 个请求。

在服务器端针对这些请求，创建 M 个对应的传输线程，每个线程负责为用户发送数据。由于 M 大小有限，因此为一个用户创建的线程较少。每个子线程需要处理的数据量较小（一个线程处理 4 幅图像），各子线程工作周期较短。

为单一用户的多个请求分别创建对应传输线程的优缺点描述如下：

- (1) 为用户的请求创建多个传输线程，可以充分利用机器资源，提高传输速率。
- (2) 第 n 层的图像几乎可以同步更新到第 $n+1$ 层，多层图像并存的时间大大缩短，给用户更好的视觉感受。
- (3) 为每个用户的请求分别创建一组传输线程^[4]会在一定程度上增加服务器端程序的工作量和复杂度。

WebGIS 服务器中针对单客户端的线程池如图 4 所示。

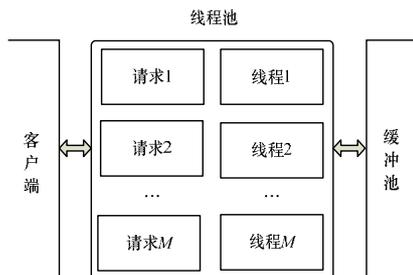


图 4 WebGIS 服务器中针对单客户端的线程池

在图 4 中，左端为单客户端，负责向服务器发送图像数据请求以及接收显示图像，其某一时刻发出的请求数为 M ；中部为服务器的线程池，请求 1, 2, ..., M 为当前客户端发送的请求，对应 M 个处理线程；右端为缓冲池，线程池中的每个线程访问缓冲池，查找各自所需的图像数据，查找成功则由该线程负责将数据传输至客户端，若查找失败，则从硬盘查找数据并进行传输，如果磁盘中也没有所需数据，则向客户端回传状态，结束该线程。

4 服务器缓冲池技术

由于在 WebGIS 数据传输中，传输的数据容量巨大，因此提高传输效率是需要解决的一大问题。笔者在 WebGIS 系统中为服务器设计了一个数据缓冲池，充分利用服务器主机的性能，减少了发生客户请求时服务器程序搜索磁盘的次数，提高了数据传输速度。

4.1 缓冲池技术

在涉及数据处理的系统中，通过判断预先将数据读入主存的方法称为缓冲池技术，主存中存储该数据集的空间称为缓冲池。该技术一般采用队列等数据结构实现。

4.2 图像数据缓冲策略

缓冲策略是缓冲池技术中最重要的问题，良好的缓冲策略会使缓冲池发挥最大作用。必须通过合理设计缓冲池的规模，实现高效缓冲和资源有效利用间的平衡。本系统在主存创建一个先入先出的队列来实现缓冲池机制。图像数据缓冲策略如图 5 所示，将第 n 层中的图像 0 关系到的图像数据导入到缓冲池中。图 5 中左上角表示第 n 层中的图像 0，在右上角的 9 个图中，中间的 0 代表第 n 层的图像 0，周围的图像 1, 2, ..., 8 表示图像 0 在该层延伸出的 8 幅图像。图 5 中左下角的 4 个图表示第 $n+1$ 层的图像 0 对应在第 $n+1$ 层的 4 幅分割图像，图像 0 的延伸图像和分割图像通过一定缓冲策略进行筛选，从磁盘导入到缓冲池中。

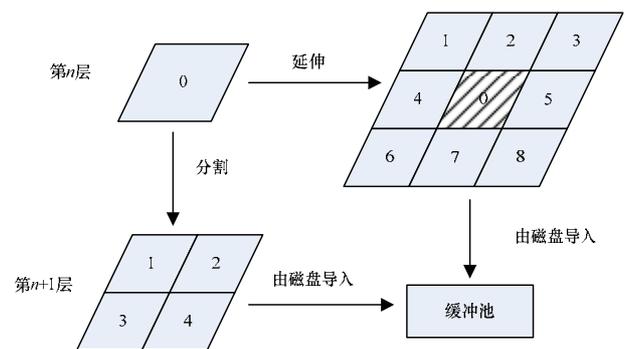


图 5 图像数据缓冲策略

图像数据缓冲策略描述如下：

(1) 假设地图数据有 0~ N 共 $N+1$ 层。

(2) 当客户端当前显示为第 n 层 ($0 \leq n \leq N-1$) 时，待缓冲数据分为 2 类，一类是对应第 $n+1$ 层的 4 幅分割图像，另一类是对应本层的延伸图像。如图 5 所示，图像 0 在本层对应的延伸数据有 8 幅图像，仅当图像 0 处于客户端屏幕边缘时，才有延伸图像。此时，位于图像客户端屏幕 4 个角上的 4 幅图像分别延伸到对应的图像 (1, 2, 4), (2, 3, 5), (4, 6, 7) 和 (5, 7, 8)；位于左、右、上、下的各幅图像分别延伸到对应的图像 (4), (5), (2) 和 (7)。

(3)当客户端当前显示为第 N 层时,图像本层的延伸图像选择策略与步骤(2)相同。

(4)将步骤(2)和步骤(3)选择出的 2 类图像送至缓冲池,如果缓冲池满,则采用先入先出算法以缓冲最新的图像数据。

经过上述 4 个步骤,对于用户在客户端进行的一般操作,所有当前图像可能延伸到的图像数据均得到了缓冲。

理论上讲,缓冲池越大,缓冲到内存中的数据就越多,服务器接收到用户请求时查找磁盘的次数越少,传输速率就越高。但服务器内存有限,无需过多浪费,且应预留内存来处理其他请求。因此,进行缓冲池规模设计时,考虑如下 2 种情况:

(1)单用户连接服务器。缓冲池中的数据仅供当前用户使用,缓冲池规模较小。在笔者进行的实验中,服务器空闲内存约 1.6 GB,每幅图像大小约为 260 KB,仅需分配 160 MB 内存即可满足上述图像数据缓冲策略的要求。

(2)多用户并发连接服务器。缓冲池被设计成多用户共享式,此时缓冲池规模较大。此实验中服务器初始状态与单一用户时基本相同。2 个用户同时连接时,约需要为缓冲池分配 18%的空闲内存,3 个用户同时连接时为 25%, N 个用户同时连接时,缓冲池需分配的内存约为 $160 \times (1+0.8)^{N-1}$ MB,可以最多分配 90%的空闲内存给缓冲池。

在实验中,缓冲池的规模设计方法可以使上述缓冲策略得以实现。在该缓冲策略下,多数请求需要的数据均在服务器缓冲池中得到了缓冲,减少了服务器接收到请求后搜索硬盘的次数,节省了大量时间。因此,该缓冲策略基本可行,可以起到优化数据传输的作用。

5 实验及结论

笔者在相同的服务器和客户端环境下,通过比较单一用户连接服务器时的传输效率,说明了本文设计的单一用户多线程机制及数据缓冲池机制在 WebGIS 数据传输中的优势。实验环境与实验结果见表 1。从实验结果可以看到,采用本文设计的单一用户多线程机制及数据缓冲池机制进行 WebGIS 数据传输时,传输效率相对于普通传输方法提高了近 30%。

(上接第 78 页)

$$x_i = 2 \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} - 1$$

归一化后 $x_i \in [-1, +1]$ 。用 SVM 分类预测精度对比结果(见表 4)。

表 4 SVM(1-V-r)分类预测实验结果精度 (%)

算法	数据集				
	iris	breast	wine	sonar	vehicle
CAIR	94.9	93.3	94.4	59.5	64.7
CAIM	90.0	94.2	94.4	76.2	51.2
本文算法	98.3	97.1	86.1	50.0	70.0

表 3 和表 4 结果显示,本文提出的算法的正确识别率和分类预测精度总体较 CAIR 和 CAIM 算法效果好。这表明该算法减少了离散造成的信息损失。

5 结束语

本文将属性重要性考虑到条件属性的离散顺序中,并引入粗糙集中的观点——知识就是区分事物的能力,提出了条

表 1 实验环境与实验结果

服务器运行环境	CPU: 双核 2.0 GHz 内存: 2.0 GB 操作系统: Windows XP SP2	
客户端运行环境	CPU: 双核 1.7 GHz 内存: 1.0 GB 操作系统: Windows XP SP2	
网络	带宽: 1 Mb/s	
实验条件	客户端屏幕显示第 N 层共 8 幅图像,服务器数据包含对应第 $N+1$ 层 8×4 幅图像,在客户端进行放大操作,查看第 $N+1$ 层图像数据,记录从开始放大操作到传输完 $N+1$ 层图像数据的传输时间(重复 5 次,取平均值)	
实验方法	采用为每一用户创建对应处理线程的数据传输方式进行传输	采用本文所设计的单一用户多线程机制及数据缓冲池机制进行传输
实验原理	建立当前客户端与服务器的连接,为该用户创建处理线程。客户端向服务器发送这 8 个数据请求时,服务器为每个请求创建一个传输子线程,各子线程在服务器的缓冲池中查找对应的图像数据(在显示当前第 N 层数据时第 $N+1$ 层数据已被缓冲到缓冲池中),并进行传输	建立当前客户端与服务器的连接,为该用户创建处理线程。客户端向服务器发送这 8 个数据请求时,服务器进行响应,并依次从硬盘查找对应图像数据进行传输
实验结果	平均传输时间: 4 013 ms	平均传输时间: 2 822 ms

6 结束语

图像数据传输是 WebGIS 中的关键技术,其传输效率直接关系到 WebGIS 性能的优劣。本文在多客户端多处理线程的基础上为单一客户端的多请求设计了多线程机制,进一步提高了服务器的工作效率。采用数据缓冲池设计了图像数据缓冲策略,充分利用服务器主存空间,使图像数据传输时的磁盘搜索次数大大减少。

参考文献

- [1] 王继周, 李成名, 付俊娥. Java 多线程技术在 WebGIS 空间数据传输中的应用[J]. 计算机应用研究, 2004, 21(2): 211-216.
- [2] Robinson S, Nagel C, Glynn J. Professional C#[M]. 3rd ed. [S. l.]: Wrox Press, 2004.
- [3] 王海涛, 裴照君, 段哲民. 基于多线程和网络传输的数据采集系统的实现[J]. 电子测量技术, 2007, 3(11): 108-110, 113.
- [4] David G B, Frank K. NASA World Wind: Open Source GIS for Mission Operations[Z]. IEEEAC Paper #1048, 2006.

编辑 陈 晖

件属性可分辨率概念,将其应用到连续属性离散化算法 CAIM 中,为属性的离散化提供了一种新思路。理论和实验结果证明了该算法的有效性。

参考文献

- [1] Kurgan L A, Cios K J. CAIM Discretization Algorithm[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(2): 145-153.
- [2] 徐 燕, 怀进鹏, 王兆其. 基于区分能力大小的启发式约简算法及其应用[J]. 计算机学报, 2003, 26(1): 97-103.
- [3] Ching J Y, Wong A K C, Chan K C C. Class-dependent Discretization for Inductive Learning from Continuous and Mixed-mode Data[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1995, 17(7): 641-651.
- [4] 李国正, 王 猛. 支持向量机导论[M]. 北京: 电子工业出版社, 2000.

编辑 任吉慧