

# 有效的 WebGIS 地图服务器场负载均衡算法

黄 颖<sup>1,2</sup>, 谢 忠<sup>1,2</sup>, 郭明强<sup>1,2</sup>

(1. 中国地质大学信息工程学院, 武汉 430074; 2. 教育部 GIS 软件与应用工程研究中心, 武汉 430074)

**摘 要:** 针对现有网络地理信息系统(WebGIS)负载均衡技术的缺陷, 分析利用集群技术解决 WebGIS 负载均衡的必要性和可行性, 提出基于地图服务器场的 WebGIS 分布式负载均衡模型, 研究地图服务器场分布式动态均衡调度算法, 建立试验床, 通过一系列的试验结果验证了该算法良好的全局网络负载性。

**关键词:** 网络地理信息系统; 地图服务器场; 负载均衡

## Effective Load-balancing Algorithm for WebGIS Based on Map Server Farm

HUANG Ying<sup>1,2</sup>, XIE Zhong<sup>1,2</sup>, GUO Ming-qiang<sup>1,2</sup>

(1. Faculty of Information & Engineering, China University of Geosciences, Wuhan 430074;

2. GIS Software and Application Project Research Center of the Educational Department, Wuhan 430074)

**【Abstract】** Aiming at deficiencies of load-balancing technology in the existing WebGIS, the paper analyzes the necessity and feasibility of cluster technology and proposes a new distributed load-balancing model based on map server farm to improve WebGIS performance. The algorithm, named distributed dynamic load-balancing scheduling algorithm, is researched as the core component in the model. A test-bed is established. A series of test results prove that the algorithm has good global load balancing performance.

**【Key words】** WebGIS; map server farm; load-balancing

WebGIS 是在 Internet 或 Intranet 网络环境下的一种兼具存储、处理、分析、显示与应用地理信息的计算机系统<sup>[1]</sup>。它的基本思想是在互联网上提供地理信息, 实现地理数据的远程互操作和互运算, 并进行在线分析和数据挖掘<sup>[2]</sup>, 让用户通过浏览器浏览和获取地理信息系统中的数据和功能。

### 1 现有 WebGIS 负载均衡技术及其缺陷

现有 WebGIS 系统主要从硬件和软件 2 个方面解决负载均衡问题。

硬件上通过专用的负载均衡器或提高地图服务器的 CPU 处理速度、增加内存容量等办法<sup>[3]</sup>提高系统性能, 但代价昂贵。

软件方法提供了廉价而有效的负载均衡机制, 目前已有相关探索, 如最短任务队列法<sup>[4]</sup>、基于网络轮询法、比例系数法<sup>[5]</sup>等。

以上算法的基本思想是通过计算当前任务大小与当前网络负载量, 最终把目标任务传送给当前网络上负载最小的一台处理器处理。

由于 WebGIS 系统自身设计的复杂性和特殊性, 使得大量的数据处理操作集中在地图服务器端。测试发现, 地图服务器将矢量地图数据转换成栅格图片所用的时间与图片从地图服务器发送到 Web 站点所用的时间相差了好几个数量级。后台的矢量数据量越大, 这种差别就越发明。随着 WebGIS 系统用户数和网络流量呈几何级数增长, 对地图服务器的可伸缩性提出了很高的要求。当大量并发数据到达时, 单一地图服务器无法完全及时处理所有的请求, 将造成应答滞后、请求丢失等, 严重地导致一些 GIS 数据包因延时而重发。因

此, 利用地图服务器集群技术将多台地图服务器组成一个地图服务器场, 通过负载均衡技术将 WebGIS 系统中大用户量、计算密集型任务分担到多台地图服务器上同时处理, 以提高系统并行处理能力势在必行。

针对以上问题, 本文提出一种有效负载均衡的 WebGIS 地图服务器场分布式动态负载均衡算法, 最大限度地减少了 WebGIS 地图服务器磁盘 I/O 操作和 CPU 指令数, 有效提高了 WebGIS 系统并行处理能力, 并解决了地图服务器单点故障问题, 为用户提供了更快、更好的 WebGIS 服务。

### 2 地图服务器场分布式动态均衡调度模型

基于地图服务调度器, 构建了地图服务器场分布式动态均衡调度模型, 如图 1 所示。消息监听器<sup>[6]</sup>监听到来自 Web 服务器的请求后, 立即通知总控模块。总控模块下包含任务求解机制, 根据地图服务器场分布式动态均衡调度算法把地图服务请求均衡发送到多台分布式部署的地图服务器节点上, 达到均衡负载的目的。处理完毕后通过动态反馈机制返回给客户层。为解决地图服务调度器的单点故障问题, 本模型提供了地图服务调度器的容错处理机制。

**基金项目:** 国家“863”计划基金资助项目“地球观测与导航技术领域专题”(06AA12Z218); 国家自然科学基金资助项目(40771165); “十一五”国家科技支撑计划基金资助项目“城市空间信息基础设施共享关键技术研究及示范”(2006BAJ15B02)

**作者简介:** 黄 颖(1981-), 女, 博士研究生, 主研方向: WebGIS, 网络计算; 谢 忠, 教授、博士生导师; 郭明强, 硕士

**收稿日期:** 2008-09-28 **E-mail:** mayying\_2003@163.com

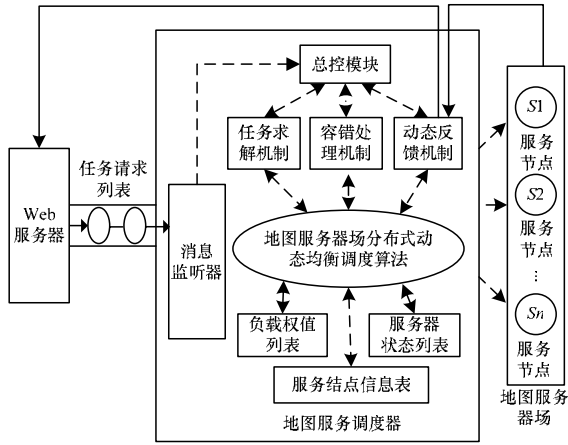


图1 地图服务器场分布式动态调度模型

### 3 算法求解及容错机制

#### 3.1 算法求解

##### 3.1.1 地图服务初始化

地图服务调度器初始化地图服务器场负载权值<sup>[7]</sup>列表、服务节点信息表及服务节点状态列表。

各地图服务器负载权值主要由以下三方面决定：地图服务器的软硬件性能线性综合参数  $H_i$ ，权值为  $W(h)$ ；地图服务调度器与地图服务器场之间的网络状况参数  $T_i$ ，权值为  $W(t)$ ；地图服务器即时处理能力  $P_i$ ，权值为  $W(p)$ 。则有：

$$W(h)+W(t)+W(p)=1 \quad (H_i, P_i, T_i < 1)$$

负载权值算法计算流程如下：

(1)根据地图服务器软硬件配置信息计算地图服务器软硬件配置所占权值。各个硬件配置的性能值和权值参数定义如表1所示。其中  $C_i, M_i, D_i, X_i, N_i, O_i < 1, W(c)+W(m)+W(d)+W(x)+W(n)+W(o)=1$ 。

表1 性能值、权值参数定义表

硬件名称	性能值	权值
CPU	$C_i$	$W(c)$
内存	$M_i$	$W(m)$
硬盘	$D_i$	$W(d)$
显卡	$X_i$	$W(x)$
网卡	$N_i$	$W(n)$
操作系统	$O_i$	$W(o)$

根据上述参数和服务器软硬件配置权值计算地图服务器软硬件性能线性综合参数：

$$H_i=(C_i \cdot W(c)+M_i \cdot W(m)+D_i \cdot W(d)+X_i \cdot W(x)+N_i \cdot W(n)+O_i \cdot W(o)) \cdot W(h) \quad (1)$$

(2)计算出软硬件性能线性综合参数后，再根据地图服务器的网络延时计算出各个地图服务器的网络状况参数，其计算公式如下：

$$T_i = \frac{W(t)}{t_i \sum_{i=1}^n 1/t_i} \quad (2)$$

其中， $t_i$  是各个地图服务器与地图服务调度器之间的网络延时； $n$  为地图服务器的数目。

(3)再向各个地图服务器提交运算任务，根据运算时间计算各个地图服务器处理能力权值，其公式如下：

$$P_i = \frac{W(p)}{(p_i - t_i) \sum_{i=1}^n 1/(p_i - t_i)} \quad (3)$$

其中， $t_i, n$  同上； $p_i$  为地图服务调度器向地图服务器提交运算任务并得到结果所花的总时间。

通过式(1)~式(3)，可得出各个地图服务器的负载权值：

$$L_i = H_i \cdot W(h) + T_i \cdot W(t) + P_i \cdot W(p) \quad (4)$$

计算出地图服务器的负载权值后，地图服务调度器启动服务，开始侦听服务端口，准备接收 Web 服务器的地图服务请求。同时启动地图服务调度器监控线程，定时监控各个地图服务器状态。

##### 3.1.2 地图服务器场分布式动态均衡调度算法的调用

当地图服务调度器接收到 Web 服务器的请求后，地图服务调度器先根据保存的地图服务器场各节点信息列表中的地图服务器地址信息检测各个地图服务器当前工作是否正常，并修改相应的服务器列表信息。根据当前工作正常的地图服务器的负载权值对地图范围进行剖分。采用地图服务器场分布式动态均衡调度算法把剖分后的地图范围发送到相应的地图服务器分别进行处理。

可以采取行平面剖分法或列平面剖分法对地图可视窗口进行剖分。下面以列平面剖分法为例阐述。

根据式(4)得到的各个地图服务器的负载权值得各个地图服务器将要处理的地图可视范围比例  $S_i$ ：

$$S_i = \frac{L_i}{\sum_{i=1}^n L_i} \quad (5)$$

设 Web 服务器发送的地图服务请求可视范围逻辑坐标  $X$  最小值为  $X_{min}$ ，最大值为  $X_{max}$ ， $Y$  最小值是  $Y_{min}$ ，最大值是  $Y_{max}$ 。地图服务器所处理的地图可视范围的逻辑坐标  $X$  最小值是  $X_{min_i}$ ， $X$  最大值是  $X_{max_i}$ 。采用列平面剖分法，则  $Y_{min_i}=Y_{min}$ ， $Y_{max_i}=Y_{max}$ 。计算公式如下所示：

$$X_{min_i} = \begin{cases} X_{min} & i = 0 \\ X_{min_i} = X_{max_{(i-1)}} & i > 0 \end{cases}$$

$$X_{max_i} = \begin{cases} X_{max} & i = n \\ X_{min_i} + X_{max} - X_{min} \cdot S_i & 0 < i < n \end{cases} \quad (6)$$

根据上述逻辑坐标就可以把地图可视范围剖分为  $n$  个部分，然后分别把各个部分发送给地图服务器场中各服务器进行处理。地图服务器负载权值越大，所得到的地图可视范围就越大，反之则越小。

##### 3.1.3 动态反馈过程

地图服务器定时收集负载权值，并以 XML 消息机制发送给地图服务调度器，地图服务调度器定时更新负载权值列表。当地图服务器处理完收到的请求后，立即把结果返回给地图服务调度器，地图服务调度器把各个地图服务器的处理结果整合在一起，然后返回给 Web 服务器。

#### 3.2 容错处理机制

尽管地图服务器场分布式动态均衡调度算法可以较好地处理地图服务请求分配给服务器场中的各个地图服务器去负载，但如果调度算法没有对网络系统状况的检测方式和能力，一旦在某台地图服务器或某台地图服务调度器与地图服务器网络间出现故障的情况下，地图服务调度器依然把一部分数据流量引向那台地图服务器，这势必造成大量的服务请求丢失，达不到不间断可用性的要求。所以良好的调度算法应有对网络故障、服务器系统故障、应用服务故障的检测方式和能力，以便一台地图服务器故障可以把数据流量引向网络连接正常的地图服务器。

地图服务调度器检测地图服务器时采用端口扫描法对地图服务器场中提供服务的地图服务器端口进行探测。具体实现方法如下：利用 SOCKET 套接字的 Connect 方法尝试连接地图服务器的服务端口(如 5141)是否可以连通，从而获得该

端口的状态。如果处于侦听状态，那么连接就能够成功，否则这个端口就不能使用，即没有提供服务。使用这种方法的优点是：无需任何权限。系统中的任何用户都有权利使用这个调用。另一个优点是速度快。可以同时打开多个 SOCKET，从而加速扫描。

地图服务调度器向地图服务器发送请求后，如果某个地图服务器出现异常或者处理结果在指定时间内没有返回，则取消该请求，然后根据已完成任务的地图服务器的负载权值  $L_i$  重新把该请求所处理的地图范围进行剖分并发送到各个地图服务器上进行处理。如此循环，直到所有的请求均已完成。

如果在处理过程中地图服务调度器出现异常以致不能继续提供服务，则 Web 服务器将根据保存的用户状态将请求重新发送到另外的工作正常的地图服务调度器，从而避免地图服务器单点故障。

#### 4 算法性能分析

在 VS2005 环境下，按照上述算法，针对用户平台多样化、系统并发要求高、访问流量大等特点，开发了基于本算法的 WebGIS 模型。使用位于高速局域网内的 4 台 PC 机构建试验床，其中 2 台 PC 机做地图服务器，1 台 PC 机做 Web 服务器，1 台 PC 机做地图服务调度器。各个测试机配置如表 2 所示。

表 2 硬件配置参数

角色	CPU	内存/GB	硬盘	网卡	显卡
地图服务器	Intel PD2.66×2	2	ST 160G/7200R	D-Link DFE-530TX	RADEON X550
地图服务器	Intel PD2.66×2	2	ST 160G/7200R	D-Link DFE-530TX	RADEON X550
Web 服务器	Intel PD2.66×2	1	ST 160G/7200R	D-Link DFE-530TX	RADEON X550
地图服务调度器	Intel PD2.66×2	1	ST 160G/7200R	D-Link DFE-530TX	RADEON X550

通过 LoadRunner 对系统进行压力负载测试。测试数据：10.53 MB 地图数据，图像大小 500×600 像素。测试分为 2 个步骤：首先进行指定用户连接数系统性能测试对比；然后对最短任务队列算法及本算法下不同连接数的服务器平均响应时间进行比较。由图 2、图 3 所示测试结果可见，最短任务队列算法 20 个用户做地图加载操作服务器平均响应时间为 25.247 s，最高为 56 s，而采用本算法 20 个用户做地图加载操作服务器平均响应时间为 16.112 s，最高为 35 s。图 4 为客户端每增加 5 个用户并发进行地图加载操作和地图放大操作时本算法与最短队列算法服务器的平均响应时间对比图。

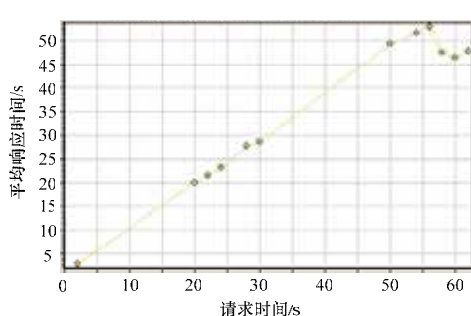


图 2 基于最短队列算法的地图加载操作

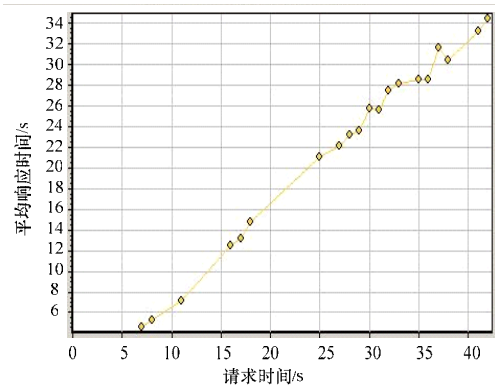


图 3 基于本算法的地图加载操作

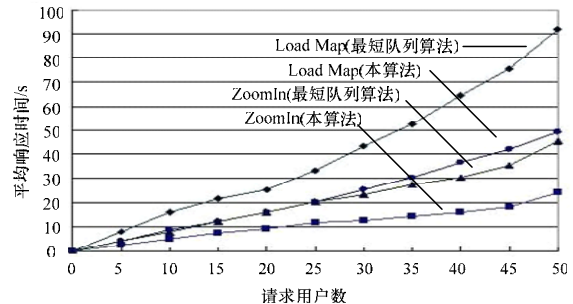


图 4 最短队列算法与本算法服务器平均响应时间对比图

由以上测试结果可见，使用本算法可以明显地缩短系统响应时间，提高地图服务器场的并发处理能力，具有较好的稳定性、并发性和抗负载能力。

#### 5 结束语

本文对 WebGIS 中地图服务器场负载均衡算法从设计到实现进行了详细的阐述，具体讨论了地图服务器场分布式动态均衡算法模型和实现方式。最后建立试验床对算法进行比较、分析，验证了本算法的可用性、高效性。下一步工作是通过内存拷贝机制实现地图服务器场间的数据同步，探讨负载均衡技术更深入的应用研究。

#### 参考文献

- [1] 何江, 刘仁义, 刘南. 基于 COM+ 的 WebGIS 系统设计与实现[J]. 浙江大学学报: 理学版, 2004, 31(6): 712-715.
- [2] 江飞, 周保群, 王惠芳. 一种有效负载均衡的分布式 WebGIS 体系结构模型[J]. 微计算机信息, 2006, 22(10): 215-218.
- [3] 朱江, 宋关福, 钟耳顺. 基于 Web Services 和 .NET 技术的新一代 WebGIS 研究[J]. 地理信息世界, 2004, 2(2): 17-20.
- [4] 张胜, 康志伟. 基于 .NET 技术的 WebGIS 系统的设计与实现[J]. 计算机工程, 2006, 32(15): 106-108.
- [5] 李文正, 郭巧, 王利, 等. Internet 服务器负载均衡的研究与实现[J]. 计算机工程, 2005, 31(6): 98-101.
- [6] Down D G, Lewis M E. Dynamic Load Balancing in Parallel Queuing Systems: Stability and Optimal Control[J]. European Journal of Operational Research, 2006, 168(6): 509-519.
- [7] Iqbal S, Carey G F. Performance Analysis of Dynamic Load Balancing Algorithms with Variable[J]. Journal of Parallel and Distributed Computing, 2005, 65(3): 934-941.