

文章编号:1671-8860(2013)09-1131-04

文献标志码:A

# 集群并发环境下大规模矢量数据负载均衡算法

郭明强<sup>1,2</sup> 谢忠<sup>1,2</sup> 黄颖<sup>2</sup>

(1 中国地质大学(武汉)信息工程学院,武汉市鲁磨路388号,430074)

(2 教育部GIS软件与应用工程研究中心,武汉市关山一路特1号,430074)

**摘要:**为了提高集群环境下网络地理信息系统(WebGIS)大规模矢量数据的并发访问性能,提出了集群并发环境下大规模矢量数据内容网格化负载均衡算法,研究了大规模矢量数据内容网格化方法,集群并发访问时内容网格的自动识别、分析、聚合、反馈算法,实现了面向任务的负载均衡。实验表明,本算法能在大规模、高强度的矢量数据提取和显示中均衡地分发请求,使集群服务器充分发挥其优势从而获得最小的请求响应时间。

**关键词:**网络地理信息系统;集群;矢量数据;网格;负载均衡

中图法分类号:P208; TP301; TP393

海量空间数据具有多分辨率、多数据集的特点,适合WebGIS的属性并适用于其访问特征和大规模访问量的集群负载均衡方法,特别是针对集群环境下如何提升大规模矢量数据并发访问性能方面的研究,在学术界还相对较少<sup>[1-9]</sup>。

文献[6]对矢量地图请求范围按比例进行分割,然后分发到各个服务器上。但当矢量数据内容分布不均匀时,这种范围分割方法将会导致任务分解的不公平。

WebGIS中的两大因素影响大规模矢量数据并发访问性能:①数据提取;②矢量图形绘制<sup>[1]</sup>。后台的矢量数据量越大,这种差别就越明显。当矢量数据量增长到一定规模时,矢量地图请求的平均响应时间根本无法满足客户端大用户量并发访问需求<sup>[2]</sup>。

本文针对这两个因素,设计了一种集群并发环境下大规模矢量数据内容网格化负载均衡算法,实现了让集群中的所有服务器能同时合作并发完成客户端提交的一个大规模矢量数据提取和显示任务,真正达到了面向任务负载均衡的目的。

## 1 内容网格化负载均衡算法

本文提出了一种可自由按需扩展的网络地图服务集群负载均衡模型(图1)。

### 1.1 矢量地图内容网格化思想

图1模型需要解决的关键问题是,如何把大规模矢量数据请求公平地分解到多个服务器上进行并发处理。本文采取对大规模矢量数据内容和客户端的请求范围进行网格化来解决该问题。

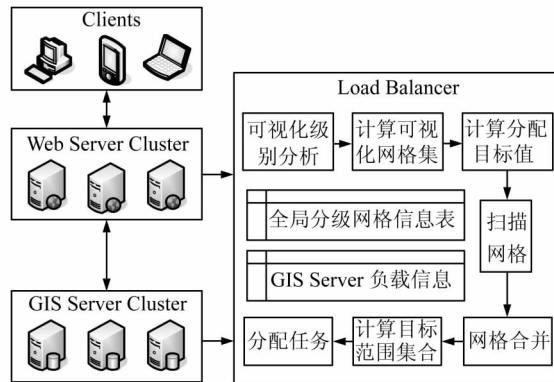


图1 网络地图服务集群负载均衡系统模型

Fig. 1 Load Balancing Model for WebGIS

#### 1) 服务器端矢量数据内容分级网格化

对服务器端矢量数据内容进行四叉树分级网格化,将网格单元格矩形范围及包含矢量要素个数等信息存储到Load Balancer中的全局分级网格信息表中。全局分级网格信息表如表1。

#### 2) 客户端可视化范围网格化

为了减小负载均衡器对所有请求的网格分析

收稿日期:2013-05-12。

项目来源:国家863计划资助项目(2009AA12Z21);国家“十二五”科技支撑计划资助项目(2011BAH06B04)。

表1 全局分级网格信息表

Tab. 1 Global Grid Information Table

字段名	类型	描述
Level	Int	网格级别
Row	Long	网格行号
Col	Long	网格列号
$Gx_{\min}$	Double	网格 x 轴最小值
$Gy_{\min}$	Double	网格 y 轴最小值
$Gx_{\max}$	Double	网格 x 轴最大值
$Gy_{\max}$	Double	网格 y 轴最大值
FeatureCount	Long	网格中要素个数

时间开销的波动,客户端窗口网格数使用经验值,设  $v$  为预设的将可视化窗口划分的网格数,用户可以根据矢量数据最大比例尺和矢量数据实际分布情况选择适当的经验值。

负载均衡器对请求范围内所有网格的矢量数据量进行分析,计算出分配给地图服务器集群中各服务器的矢量数据范围并将其分发给各个服务器,如图 2 所示。

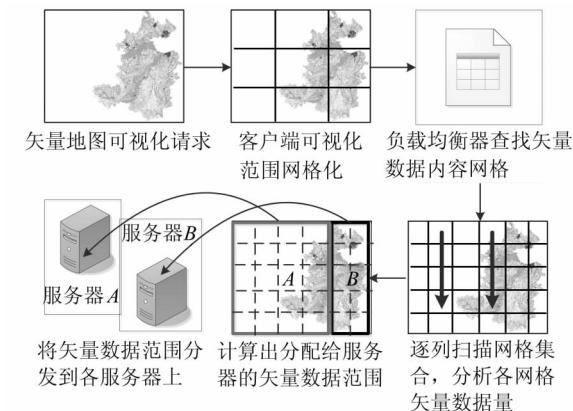


图2 矢量数据可视化请求网格化分析流程

Fig. 2 Grid Analysis Process of Vector Data Visualization

## 1.2 矢量数据内容网格化负载均衡算法

### 1) 计算矢量数据内容网格化信息。

定义 1 矢量数据范围。设矢量图层中要素分布的外包矩形范围为  $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ 。

定义 2 矢量数据内容网格数。对矢量数据外包矩形范围进行网格化,本文采用四叉树分级网格化的方法,则第  $i$  级内容网格数为  $2^i \times 2^i$ 。

定义 3 矢量数据内容网格原点。本文使用矢量数据范围左下角即  $(x_{\min}, y_{\min})$  为网格坐标轴原点。

由于本文采用四叉树网格化方法,为了方便进行网格化,把矢量数据的外包矩形范围矫正为一个正方形范围作为矢量数据内容网格的矢量数据范围。其方法是以  $(x_{\min}, y_{\min})$  为原点,并以  $\max((x_{\max} - x_{\min}), (y_{\max} - y_{\min}))$  作为正方形范围

的边长,则正方形右上角坐标为  $(x_{\min} + \max((x_{\max} - x_{\min}), (y_{\max} - y_{\min})), y_{\min} + \max((x_{\max} - x_{\min}), (y_{\max} - y_{\min}))$ 。

定义 4 矢量数据内容网格单元逻辑范围跨度  $d_i$ 。 $d_i$  是第  $i$  级网格层中一个网格的逻辑范围跨度:

$$d_i = \frac{\max((x_{\max} - x_{\min}), (y_{\max} - y_{\min}))}{2^i} \quad (1)$$

定义 5 设  $RG(x, y, i)$  为第  $i$  级第  $x$  列第  $y$  行网格中分布的矢量数据量。

为了查询网格中的矢量要素个数,必须先计算出网格包含的矢量数据的范围。根据该网格中矢量数据范围,可以从大规模矢量数据中查询到该范围内分布的矢量数据量  $RG(x, y, i)$ ,将其和网格中的矢量数据范围一起存储到 Load Balancer 端的全局分级网格信息表中。

### 2) 计算服务器负载权值。

第  $n$  个服务器负载权值  $P_n$ 。为了对请求范围中的矢量数据进行网格化分析,根据文献 [7] 中的方法计算集群中各服务器负载权值。

3) 接收客户端请求,计算客户端请求范围网格中单元格逻辑范围跨度  $d_w$ 。

设客户端当前可视化窗口高宽为  $W, H$ ,可视化窗口中矢量数据的逻辑范围为  $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ 。 $v$  为预设的将可视化窗口划分的网格数,可视化窗口网格化后网格单元格的宽度为:

$$d_w = \frac{\max((x_{\max} - x_{\min}), (y_{\max} - y_{\min}))}{v} \quad (2)$$

### 4) 根据 $d_w$ 查找服务器端内容网格级别 $i$ 。

根据  $d_w$  找到最接近的  $d_i$  (定义 4),从而找到矢量数据内容网格的级别  $i$ 。

5) 根据第  $i$  级内容网格信息计算当前客户端请求范围对应的网格集合  $GSet$ 。

定义 6 矢量数据内容网格原点为  $(x_{\min}, y_{\min})$ ,  $GSet$  开始行列计算公式如下:

$$\text{start}_c = \frac{x_{\min} - X_{\min}}{d_i} \quad (3)$$

$$\text{end}_c = \frac{x_{\max} - X_{\min}}{d_i} \quad (4)$$

$$\text{start}_r = \frac{y_{\min} - Y_{\min}}{d_i} \quad (5)$$

$$\text{end}_r = \frac{y_{\max} - Y_{\min}}{d_i} \quad (6)$$

### 6) 计算 $GSet$ 中矢量数据总量。

设第  $i$  级第  $x$  列第  $y$  行网格与请求范围的交集包含的矢量数据容量为  $G(x, y, i)$ ,从网格信息

表中查询到的该网格包含的矢量数据量为  $RG(x, y, i)$ , 设该网格与请求范围的交集为  $[Tx_{\min_x}, Tx_{\max_x}, Ty_{\min_y}, Ty_{\max_y}]$ 。根据交集与网格单元格的面积比和  $RG(x, y, i)$ , 计算出该交集内分配的矢量数据:

$$G(x, y, i) = RG(x, y, i) \times \frac{(Tx_{\max_x} - Tx_{\min_x})(Ty_{\max_y} - Ty_{\min_y})}{d_i^2} \quad (7)$$

定义 7 请求范围内矢量数据总量  $C_{\text{Total}}$ :

$$C_{\text{Total}} = \sum_{y=\text{start}_r}^{\text{end}_r} \sum_{x=\text{start}_c}^{\text{end}_c} G(x, y, i) \quad (8)$$

7) 计算集群中各服务器的矢量数据量分配目标。

定义 8 服务器目标分配矢量数据量  $C_n$ :

$$C_n = C_{\text{Total}} \times P_n \quad (9)$$

8) 扫描网格集合  $GSet$ , 计算分配给各服务器的矢量数据范围。

根据  $P_n$  扫描分析计算各地图服务器的目标分配网格集合。假设地图服务器个数为  $N$ , 算法最终的目标是将请求范围分为  $N$  份, 每个范围包含的矢量数据内容大小占请求范围内总矢量数据大小的比重等于  $P_n$ , 设服务器  $n$  分配得到的范围区间为  $[Sx_{\min_n}, Sx_{\max_n}, Sy_{\min_n}, Sy_{\max_n}]$ , 为了方便范围划分和图像结果合成, 本算法中的网格扫描采用逐列扫描方法, 最终分配给集群中各服务器的矢量数据范围  $Y$  轴的最小值与最大值与可视化范围相同, 所以每个服务器分配得到的范围为  $Sy_{\min_n} = y_{\min}, Sy_{\max_n} = y_{\max}$ 。

第  $i$  级第  $x$  列网格覆盖请求范围的矢量数据总量计算公式如下:

$$CG(x, i) = \sum_{y=\text{start}_r}^{\text{end}_r} G(x, y, i) \quad (10)$$

使用逐列扫描法逐列扫描网格, 每扫描一列(设当前扫描列为  $c$ , 当前正在为第  $h$  台服务器进行扫描), 统计目前扫描过的所有网格列的矢量数据量  $CG = \sum_{x=\text{start}_c}^c CG(x, i)$ , 如果小于  $\sum_{n=0}^h C_n$ , 则继续扫描; 如果大于  $\sum_{n=0}^h C_n$ , 则需要计算出超出目标值  $\sum_{n=0}^h C_n$  的值。对当前列进行拆分, 左面一部分给地图服务器  $n$ , 右面部分给地图服务器  $n+1$ , 右面部分占的比例为  $PL = \frac{CG - \sum_{n=0}^h C_n}{CG(c, i)}$ , 地图服务器  $n$  分配到的范围计算公式如下:

$$Sx_{\min_n} = \begin{cases} Sx_{\max_{(n-1)}}, & n > 0 \\ x_{\min}, & n = 0 \end{cases} \quad (11)$$

$$\begin{cases} X_{\min} + cd_i + (1 - PL)d_i, & n < N - 1 \\ x_{\max}, & n = N - 1 \end{cases} \quad (12)$$

$$Sy_{\min_n} = y_{\min} \quad (13)$$

$$Sy_{\max}(n) = y_{\max} \quad (14)$$

循环扫描, 直到遍历完请求范围内所有网格列, 即可得到各个地图服务器分配到的矩形范围, 然后分发给各个地图服务器, 当各个地图服务器返回结果时对结果进行合成, 返回客户端显示。

## 2 实验设计与结果

### 2.1 实验参数

本文使用位于高速局域网内的刀片中心构建试验床, 采用大规模矢量数据对本文算法进行实验, 并与传统负载均衡算法进行对比。表 2 为系统仿真参数。

表 2 系统仿真参数

Tab. 2 Simulation Parameters of System

仿真参数内容	
仿真算法	内容网格化负载均衡算法、轮询算法、最少连接算法、最小负载算法
客户端	5 个千兆以太网, 50 个客户端, 64 口以太网交换机连接
Web Server	足够处理能力, 不会成为转发瓶颈
Load Balancer	1 台 CPU Intel® Xeon® E5620 2.40 GHz (4 核, 8 线程), 8GB 内存服务器
GIS Server Cluster	4 台 CPU: Intel® Xeon® E5620 2.40 GHz (4 核, 8 线程), 8GB 内存服务器
矢量数据量	实际要素个数: 10 002 142
仿真时间	5 h
仿真统计参数	使用不同的矢量数据量模拟客户端的地图请求, 使用 50 个并发用户进行压力测试, 对比采用各种算法的矢量数据请求平均响应时间

### 2.2 矢量数据请求平均响应时间

图 3 为在 50 个用户并发访问场景下, 本算法与其他几种算法在请求不同规模大小的矢量数据量时服务器的平均响应时间的对比。由实验结果可以看出, 与传统的负载均衡算法相比, 随着矢量数据量的增大, 当矢量数据量达到 10 万时, 本文中的算法优势开始体现; 当矢量数据量达到 100 万时, 本算法的优势更加明显。在网络地图可视化应用中, 客户端发送地图数据请求时, 特别是在浏览大规模矢量数据时, 平均响应时间越小, 用户体验越好<sup>[8]</sup>。本文提出的基于矢量数据内容网格化负载均衡算法在大规模矢量数据应用环境下, 可以更好地缩短地图服务器集群的平均响应时间, 当矢量数据量增大时, 地图服务器平均响应时间平稳增长, 具有较好的稳定性、并发性和抗高负载能力。

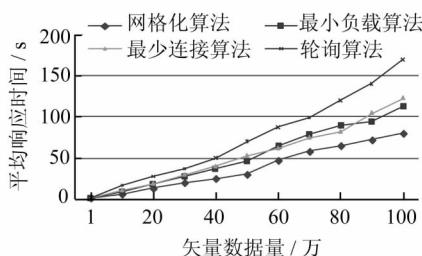


图3 矢量数据请求平均响应时间

Fig. 3 Average Response Time of Vector Data

### 3 结语

本文对集群环境下大规模矢量数据的负载均衡进行了研究,考虑到矢量数据的实际内容分布情况,设计了基于矢量数据内容网格化的负载均衡算法,将大规模的矢量数据可视化请求内容公平地分配到集群中的各个地图服务器,以提高地图服务器集群的并发处理能力,缩短平均响应时间。使用不同数据量大小的矢量数据对算法进行了验证测试,实验结果表明,在大规模矢量数据应用环境下,本文提出的负载均衡算法能更公平地实现任务分配,缩短地图服务器集群的平均响应时间。

### 参 考 文 献

[1] 王浩,喻占武,李锐等.基于开销代价的网络地理信

- 息服务负载均衡算法研究[J].测绘学报,2009,38(3):242-249
- [2] 黄颖,谢忠,吴亮,等.基于聚类调度负载均衡的WebGIS模型[J].地球科学——中国地质大学学报,2010,35(3):407-414
- [3] 黄颖,郭明强,罗显刚,等. WebGIS中GIS服务器负载均衡研究与实现[J].测绘科学,2009,34(1):182-183
- [4] 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
- [5] 李忠民,喻占武,朱莉. 基于空间数据内容的动态负载均衡方法[J].武汉大学学报·信息科学版,2009,34(5):622-624
- [6] 朱江,张立立,曾志明,等. WebGIS服务器场的负载平衡算法设计[J].计算机工程,2006,32(9):94-95
- [7] 郭明强,黄颖,谢忠. 一种基于服务器场的分布式WebGIS计算模型设计与实现[J].地理与地理信息科学,2008,24(6): 12-15
- [8] 朱莉,沈未名,李锐,等. 利用遗传算法的网络GIS集群服务器动态负载均衡算法[J].武汉大学学报·信息科学版,2011,36(6):722-725
- [9] 喻占武,李忠民,郑胜. 基于对象存储的新型网络GIS体系结构研究[J].武汉大学学报·信息科学版,2008,33(3):285-288

第一作者简介:郭明强,博士,研究方向为WebGIS。

E-mail: gmqandjxs@163.com

## Content Grid Load Balancing Algorithm for Large-Scale Vector Data in the Server Cluster Concurrent Environment

GUO Mingqiang<sup>1,2</sup> XIE Zhong<sup>1,2</sup> HUANG Ying<sup>2</sup>

(1) Faculty of Information & Engineering, China University of Geosciences(Wuhan), 388 Luomo Road, Wuhan 430074, China)

(2) GIS Software and Application Project Research Center of the Educational Department, 1 Guanshan Avenue, Wuhan 430074, China)

**Abstract:** In order to improve the concurrent access performance with large-scale vector data in WebGIS, a content grid load balancing algorithm is proposed. The server processing capability, service contents, and request time are taken into account. A proposed method to divide large-scale vector data into a content grid, the algorithm of content automatic identification, analysis, aggregation and feedback in the server cluster concurrent environment is discussed. This algorithm implements all servers in the cluster to complete visualization tasks submitted by clients at the same time and realizes task-oriented load balancing. For the extraction and display of large-scale and high-intensity vector data, the algorithm balances the servers' load efficiently and responds to requests in minimal time. As compared to the traditional load balancing algorithms, the algorithm proposed in this paper has the best performance. The larger the scale, the more obvious is the load balancing effect.

**Key words:** WebGIS; cluster; vector data; grid; load balancing

About the first author: GUO Mingqiang, Ph. D. He majors in WebGIS.

E-mail: gmqandjxs@163.com