

doi:10.3799/dqkx.2010.048

基于聚类调度负载均衡的 WebGIS 模型

黄颖^{1,2}, 谢忠^{1,2}, 吴亮^{1,2}, 郭明强^{1,2}

1. 中国地质大学信息工程学院, 湖北武汉 430074

2. 地理信息系统软件及应用教育部工程中心, 湖北武汉 430074

摘要: 为了提高网络地理信息系统(geographic information system, GIS)并发访问性能, 研究了现有 WebGIS 负载均衡技术的缺陷, 分析了在大用户量并发访问情况下利用协作任务聚类技术解决 WebGIS 负载均衡的必要性和可行性, 提出了基于聚类调度负载均衡的 WebGIS 模型, 重点研究了聚类调度负载均衡算法, 最后建立试验床, 通过一系列的试验结果验证了本模型良好的并发性。

关键词: 网络地理信息系统; 负载均衡; 聚类调度。

中图分类号: TP301.6

文章编号: 1000-2383(2010)03-0407-08

收稿日期: 2010-01-15

A WebGIS Model Based on Cluster Scheduling Load-Balancing Algorithm

HUANG Ying^{1,2}, XIE Zhong^{1,2}, WU Liang^{1,2}, GUO Ming-qiang^{1,2}

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

2. Engineering Research Center of GIS Software and Application, Ministry of Education, Wuhan 430074, China

Abstract: In order to improve the concurrent access performance of WebGIS, this paper studies deficiencies of load-balancing technologies in the current WebGIS and analyzes the necessity and feasibility to solve load-balancing in WebGIS based on collaborative tasks clustering technology under the situation of large volume of concurrent accesses. A WebGIS model based on cluster scheduling load-balancing algorithm is proposed. The key is to research the cluster scheduling load-balancing algorithm. A test-bed is established. A series of test results prove that the model has good global load balancing performance.

Key words: WebGIS; load-balancing; cluster scheduling.

0 引言

WebGIS 是互联网技术应用于地理信息系统(geographic information system, GIS)开发的产物, 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。随着网络技术和 GIS 技术的发展, WebGIS 已成为空间信息发布和共享的主要途径, 在跨平台、分布式的数据共享和互操作领域的需求日益强烈(何江等, 2004)。

在 WebGIS 环境中, 任务调度系统是其重要的组成部分, 空间计算已从单任务向顺序性、并行性、协作性任务发展。大型协作性空间计算往往被分解

为多个计算子任务, 分解后的各个子任务之间依赖关系对 WebGIS 中大型协作性空间计算任务调度和负载均衡提出了新的挑战。目前主流的一些 WebGIS 系统, 如 ArcGIS Server、MapGuide Enterprise、GeoBeans 等在某些方面具有良好的表现, 但仍存在网络传输和地图服务器负担过重、并发访问能力不适应日益扩大的 WebGIS 系统规模的不足, 该缺陷制约了 WebGIS 的应用。

本文通过分析现有 WebGIS 负载均衡技术的缺陷, 提出一种有效的基于聚类调度负载均衡的 WebGIS 模型, 有效提高了 WebGIS 系统并行处理能力, 并建立试验床, 通过一系列的试验结果验证了本模型良好的全局负载性。

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(江飞等, 2006)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、基于网络轮询法、比例系数法等。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。

目前已有的对 WebGIS 中负载均衡的研究大多是针对单任务、顺序性任务。单任务、顺序性任务是 WebGIS 中空间计算任务的简单形式, 解决的问题比较简单且未能体现任务之间的数据关联与优先约束关系; 而传统的有向无环图 (directed acyclic graph, DAG) 任务图只刻画了任务的计算量、通信量与优先关系, 只能作为同构计算环境的调度模型(张胜和康志伟, 2006), 在为协作性任务选取资源时需要综合考虑任务的计算量、任务之间的依赖关系及资源的负载能力、资源之间的网络状况。显然, 一个能够综合描述上述各种因素的调度模型和基于该模型的高效的求解方法是协作性任务调度的关键。

2 基于聚类调度负载均衡的 WebGIS 模型

网络地理信息服务中的 3 大类问题使得用户请求 WebGIS 系统响应的时间延长。一是数据提取; 二是数据传输; 三是数据显示(王浩等, 2009)。测试发现: 地图服务器在处理数据加载和空间分析操作时所用的时间与处理结果从地图服务器传输到 Web 站点显示所用的时间相差了好几个数量级。后台的数据量越大, 这种差别就越明显。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并发性能提出了很高的要求。因此, 并发访问能力和响应速度一直是 WebGIS 模型的主要改进方向。

WebGIS 的体系结构经历了从集中式体系结构和分布式两层体系结构到分布式多层体系结构的演变(王玉娥和刘晖, 2009)。目前主流的分布式多层 WebGIS 模型主要有以下两种: (1) 基于 Web 服务 + 组件的 WebGIS 模型(张胜和康志伟, 2006; 李光辉等, 2008), 如图 1a 所示; (2) 基于 Socket + 组件的

WebGIS 模型(郭明强等, 2008; 谭靖等, 2008; 周泽兵和边馥苓, 2006), 如图 1b 所示。

张胜和康志伟(2006)、李光辉等(2008)均采用 Web Service 直接调用 GIS 组件来实现网络地理信息服务。Web Service 可以供网上任何能够发现它的应用程序调用, 甚至其他的 Web Service 也可以调用它。利用 Web Service, 可以很好地实现服务在 Web 层次上的相互操作(罗显刚等, 2006)。但是, 通常的 Web Service 调用是一种静态引用的调用过程, 在跨越网络防火墙时, 大量的数据传输和分布式数据处理导致工作效率不是很理想(吴信才和吴亮, 2006)。另外, Web Service 依赖于互联网信息服务 (internet information services, IIS) 或其他 Web Service, 性能受到极大的限制, 在进行大型的空间计算任务时, 很难支持大用户量并发访问, 服务器的并发处理能力较弱。目前的 WebGIS 系统中 Web 服务器与地图服务器间紧密耦合, 不同平台的服务器进行部署需要复杂的接口完成交互, 系统的集成和维护代价较大。张胜和康志伟(2006)的研究中服务组件可以通过一个配置文件“集成”到应用服务器框架中实现服务的组合; 李光辉等(2008)通过一定的负载均衡算法将地图服务器“附加”进 WebGIS 系统。以上方法都部分地实现了业务逻辑和组件的重用性。但其所设计的地图服务器的功能均不可扩展, 若需要在 WebGIS 系统中扩展一些企业级应用功能则显得力不从心。

郭明强等(2008)、谭靖等(2008)以及周泽兵和边馥苓(2006)均采用了基于 socket 的通信机制。虽然基于 socket 的通信机制能够快速处理客户端提出的各种请求, 但在请求前必须与地图服务器端的 socket 监控程序建立连接通路。在实际应用中, 客户端与服务器端将一直保持 socket 连接, 因此, 在 WebGIS 客户端必须通过 session 保存 socket 连接, 当访问用户量较大时, 过多的创建 socket 通信连接将占用 WebGIS 客户端和地图服务器端大量资源, 极大地限制了 WebGIS 系统的并发处理能力。

为了解决以上问题, 本文提出了一种有效的聚类调度负载均衡 WebGIS 模型。摒弃了 socket 通信以及 Web 服务与组件紧密耦合的方式, 整个体系采用 Web 服务 + 负载均衡调度器 + 地图服务器的机制, 所有的 GIS 应用功能均由独立的地图服务器上的空间计算任务完成, 用户在使用过程中直接调用服务, 然后由服务发送操作请求到负载均衡调度器, 通过聚类调度负载均衡算法分配给相应的地图服务

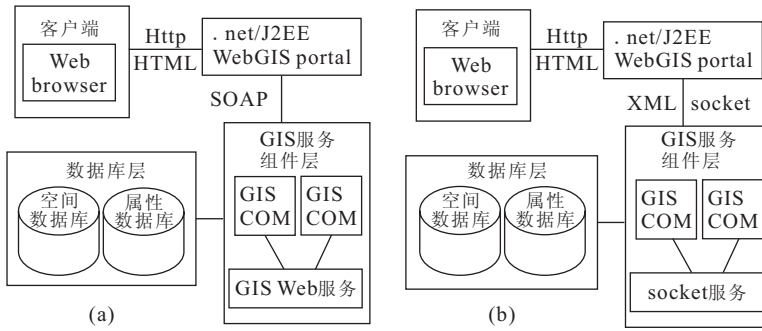


图 1 主流分布式多层 WebGIS 模型

Fig. 1 mainstream distributed multi-tier WebGIS model

a. 基于 Web 服务+组件的 WebGIS 模型; b. 基于 Socket+组件的 WebGIS 模型

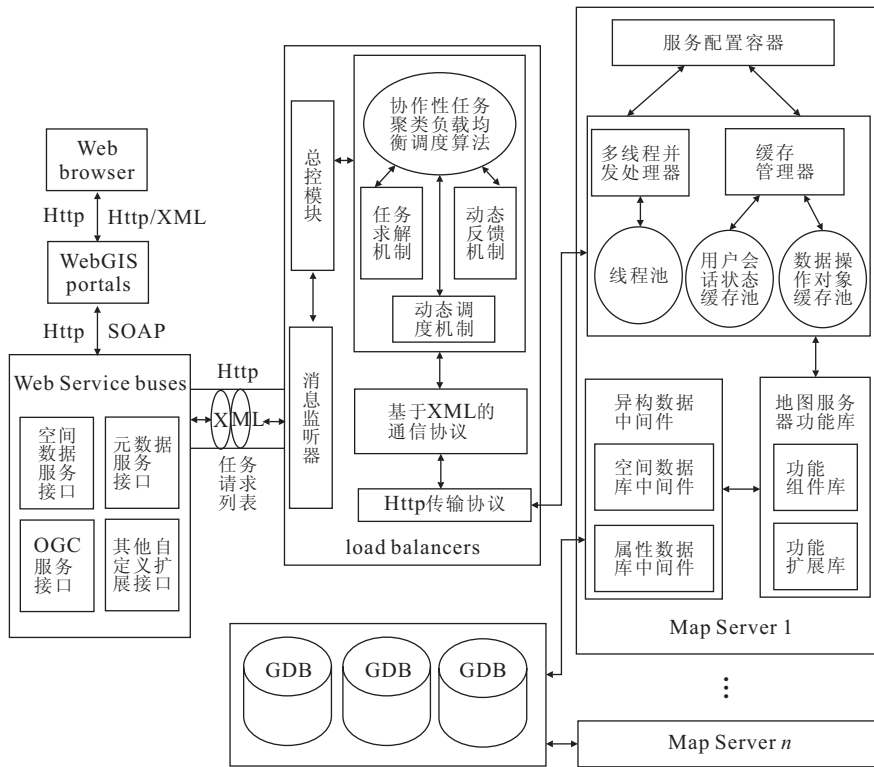


图 2 基于聚类调度负载均衡的 WebGIS 模型

Fig. 2 The cluster scheduling load-balancing WebGIS model

器处理,地图服务器多线程并发处理完成任务的并发操作,有效提高了 WebGIS 系统的并发能力、容错能力及可伸缩性.模型结构如图 2 所示.

本模型由客户端(Web browser)、WebGIS 门户(WebGIS portals)、Web 服务总线(Web service buses)、负载均衡调度器(load blancers)和地图服务器(map servers)5 个层次组成,层间结构清晰,耦合度低,相互联系形成一个有机整体,较大地提升了 WebGIS 系统性能.各层通过分布式部署,使模型具有可伸缩性,易于维护和配置,降低了使用成本.

2.1 客户端与 WebGIS 门户

客户端采用普通的 Web 浏览器浏览 WebGIS 门户网站. WebGIS 门户网站通过 IIS 或 Tomcat 等网络信息服务器,把来自客户的请求以 SOAP 信息包的形式通过 Http 协议发送给 Web 服务总线相应接口,实现客户端 GIS 计算.

2.2 Web 服务总线层

以统一的分布式 Web 服务接口的形式向用户提供集成和共享的 GIS 服务相关功能,并通过 Http 协议向负载均衡调度器发送以 XML 描述的任务请

求列表。Web 服务总线包括 4 类服务接口:空间数据服务接口、元数据服务接口、OGC 服务接口和其他自定义扩展接口。

2.3 负载均衡调度器层

通过聚类调度负载均衡算法,合理分配来自 Web 服务总线的 XML 处理请求,解决多个地图服务器的负载均衡问题。消息监听器监听到来自 Web 服务总线的请求后,立即通知总控模块。总控模块下包含任务求解机制,根据聚类调度负载均衡调度算法把来自 Web 服务总线的 XML 处理请求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。为解决负载均衡调度器的单点故障问题,本模型提供了负载均衡调度器的容错处理机制。

2.4 地图服务器层

处理负载均衡器分配的地图服务处理请求,并把结果返回给负载均衡器聚合处理,最后呈现给客户层,为用户提供高效的地图服务。采用组件、多线程、缓存等技术提供栅格和矢量数据相关服务。

服务配置容器:包括多线程并发处理器及缓存管理器。

多线程并发处理器:负载均衡调度器主要负责多个单用户请求的任务分解,而多线程并发处理器解决多地图服务器多任务并发问题。两者相辅相成,可大幅度提高整个 WebGIS 模型的并发处理性能。

缓存管理器:包括用户会话状态缓存池及数据操作对象缓存池。(1)用户会话状态缓存池:保存用户访问的地图状态对象,以保证系统出现故障时服务器上所有用户的请求状态不丢失,以便用户继续透明请求地图服务;(2)数据操作对象缓存池:可将用户已创建的数据对象缓存,以便重复使用,从而在一定程度上减少频繁创建对象所造成的开销。

地图服务器功能库:包括功能组件库与功能扩展库。(1)功能组件库:采用组件式的开发思想,以接口的形式提供各种 GIS 功能,是 GIS 应用的最终实现者,每个功能组件根据不同粒度实现不同 GIS 功能,多个功能组件可根据需要动态组成一个新的应用,实现组件的复用;(2)功能扩展库:提供可定制的 GIS 功能组件,开发人员可根据系统需求将个性化的功能组件嵌入到地图服务器功能库中,降低开发成本,加快开发进度,提供系统的灵活性和扩展性。同时提供系统运行状态报告。

异构数据中间件:包括异构空间数据库中间件和异构属性数据库中间件,用于操作多源异构空间

数据和属性数据。

2.5 数据层

包括栅格数据库、矢量数据库和属性数据库,保存 WebGIS 系统所需的各种数据信息。

3 聚类调度负载均衡机制

3.1 负载均衡算法

3.1.1 动态计算负载均衡权值 当地图服务器结点初次加入到地图服务器集群中使用,根据结点的软硬件配置情况对每个结点都设定一个初始负载权值(通常硬件配置越高的结点默认值越低),在负载均衡器上也先使用这个权值初始化地图服务器负载权值列表、地图服务器结点信息表及服务器状态列表。然后,随着结点负载的变化,负载均衡调度器对权值进行动态调整。

结合每个结点当前的权值,可以计算出新的权值。动态权值的目的是要正确反映结点负载的状况,以预测结点将来可能的负载变化。对于不同类型的系统应用,各个参数的重要程度也有所不同。典型的 WebGIS 应用环境下,可用处理器、内存资源和当前处理能力(服务响应时间)就非常重要;如果用户以长事务为主,则 CPU 使用率和可用内存就相对重要一些。为了方便在系统运行过程中针对不同的应用对各个参数的比例进行适当调整,笔者为每一个参数设定一个常量系数,用来表示各个负载参数的重要程度。各地图服务器负载权值主要由以下 3 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 $W(h)$;负载均衡调度器与地图服务器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 $W(p)$ 。

动态计算负载权值流程如下:

首先,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定义如表 1 所示。

根据表 1 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 :

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

式(1)中各个动态性能参数依次表示为: $W(c)$ 为 CPU 使用率; $W(m)$ 为内存使用率; $W(d)$ 和 $W(o)$ 为磁盘 I/O 访问率; $W(n)$ 为网络带宽使用率。

其次,计算出软硬件性能线性综合参数后,再根

表 1 性能值、权值参数定义

Table 1 The definition of performance value and weight param

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

注: $C_i, M_i, D_i, N_i, O_i < 1; W(c) + W(m) + W(d) + W(n) + W(o) = 1$.

据地图服务器的网络延时 t_i , 计算出各个地图服务器的网络状况参数, 其计算公式如下:

$$T_i = t_i \cdot W(t) \quad (2)$$

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

$$P_i = p_i \cdot W(p) \quad (3)$$

通过式(1)~(3), 可得出各个地图服务器的动态负载权值:

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

负载均衡调度器周期性的运行动态采集程序, 查询该结点的各负载参数, 并计算出动态权值 L_i .

3.1.2 生成协作性空间计算协作聚类 当负载均衡调度器接收到 Web 服务器发送的一个协作性空间计算任务请求时, 从协作空间计算任务描述器得到空间计算任务的非循环有向图, 即各子任务执行的时序关系及各个子任务之间的依赖关系, 根据各个子任务之间的通信时间和计算时间把子任务构成协作任务聚类. 设得到任务图(图 3). 任务图描述如下:

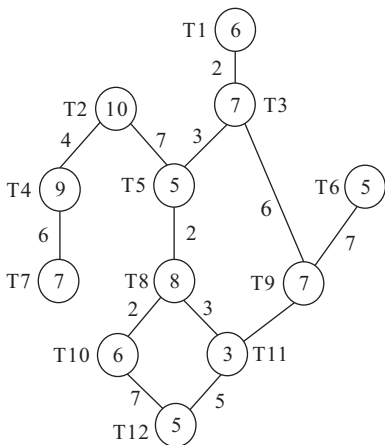


图 3 协作性计算任务

Fig. 3 The collaborative computing tasks

$$G = (T, E, e, c);$$

$$T = \{t_1, t_2, \dots, t_n\} n \text{ 个子任务};$$

$$E = \{(t_i, t_j) \mid t_i, t_j \text{ 之间需要进行通信}\};$$

$$e(t_i, t_j) \text{ 的计算时间};$$

$$c(t_i, t_j) \text{ 的通信时间}.$$

在考虑通信时间时, 硬件统一时钟, 同样性能的任何两机之间, 相同通信量的通信代价一样. 计算时间是根据协作性空间计算任务描述器中各个子任务对网络和服务器资源的需求信息等计算得到的.

为了提高系统的总体效率, 执行任务复制, 在多个服务节点上同时有同一个任务的副本, 这样因为减少了子任务的通信时间而使整个空间计算任务提前完成.

本调度策略需要计算出 G 中每个子任务的最早可能的开始时间, 如果该调度使每个子任务均能在最早可能开始的时间开始执行, 则该调度方案是最优的. 下列算法用于求某一任务 v 的最早开始时间 $s(v)$.

贪心调度函数:

设聚类 $C = \{v\} \cup \{\mu_1, \mu_2, \dots, \mu_k\}$, 其中, $\mu_1, \mu_2, \dots, \mu_k$ 是 v 的所有父任务结点, 显然, v 的最早开始时间不会早于 $\{\mu_1, \mu_2, \dots, \mu_k\}$ 的最早完成时间:

定义为 $f\{\mu_1, \mu_2, \dots, \mu_k\}$;

最大弦值函数:

$$\max g(C) = \max\{g(u, w) \mid u \in V - C, w \in C, (u, w) \in E\} \quad (5)$$

式(5)中, $g(u, w)$ 为边 (u, w) 的弦值, $g(u, w) = s(u) + e(u) + c(u, w)$, 即: 边 (u, w) 的弦值为 u 的最早开始时间, u 的耗时和 u 与 w 的通信时间之和, 这个时间可能就是 w 的最早开始时间. 因为 w 的前导任务结点中还可能存在迫使 w 晚开始的任务结点.

显然, 在一个给定的调度中, 对于 $\forall v \in V, s(v) \geq \max\{f\{\mu_1, \mu_2, \dots, \mu_k\}, \max g(C)\}$, 其中, C 为 v 和 v 的所有父任务结点的集合: $C = \{v, \mu_1, \mu_2, \dots, \mu_k\}$. 如图 4 所示.

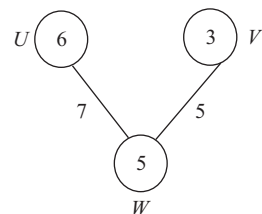


图 4 子任务最早开始时间

Fig. 4 The earliest start time of sub tasks

$g(u, w) = 13, g(v, w) = 8$, 则 $s(w) = 13$.

算法流程:

① $C = \{v\}$;

② $var-g = \max g(C)$; 取 v 的最早可能开始时间的初值;

③ $var-s = var-g$;

重复下列步骤, 直到从中跳出

④ 从 C 和 $V-C$ 中选出 $w \in C$ 和 $u \in V-C$, 使得 $g(u, w) = var-g$;

⑤ $C = C \cup \{u\}$;

⑥ $var-f = f(C - \{v\})$;

⑦ $var-g = \max g(C)$;

⑧ 如果 $var-s < \max\{var-f, var-g\}$ 则: $C = C - \{u\}$; 转 4; 表示 u 不能加入 C , 否则 $var-s = \max\{var-f, var-g\}$;

⑨ 返回 $var-s$.

上述算法在计算给定任务结点 v 的最早开始时间的同时, 计算出 v 的协作任务聚类 C , C 中的任务将被分配到同一个空间计算服务结点进行处理. 从协作任务图中的根结点开始, 逐层地对每个顶点 v 计算其 $C(v)$, 最后考虑如果 $C(v) \subseteq C(w)$, 则只用保留 $C(w)$, 这样图 4 最后得到的协作任务聚类如图 5 所示.

3.1.3 分配协作任务聚类 得到上述协作任务聚类 C_1, C_2, C_3 和 C_4 后, 根据各个协作任务聚类的计算成本 $S(i)$, 把它们进行排序, 然后保存到协作任务聚类列表, 以供负载均衡调度器使用.

当负载均衡调度器接收到任务请求后, 负载均衡调度器先根据保存的地图服务器各结点信息列表

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. 根据协作任务聚类列表中各个协作任务聚类的计算成本和各个空间计算服务节点当前的负载权值, 对各个协作任务聚类进行分配, 服务结点负载权值越大, 所得到的任务聚类的计算成本就越小, 反之则越大, 从而达到减少协作性空间计算任务的完成时间和平衡空间计算节点负载的目的.

3.1.4 动态反馈过程 地图服务器定时收集负载权值, 并以 XML 消息机制发送给负载均衡调度器, 负载均衡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, 立即把结果返回给负载均衡调度器, 负载均衡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 容错机制

(1) 在功能组件层, 对各种数据操作均进行了异常处理, 以保正系统正常运行, 当操作失败时, 功能组件层把错误信息通知给地图服务器, 地图服务器再把错误消息发送给 Web 服务总线层, Web 服务总线层根据结果的各种情况组织反馈结果发送给客户端.

(2) 负载均衡器与地图服务器之间的通信也进行了容错处理, 如果在系统运行过程中, 地图服务器重启或者由于其他原因而停止服务, 用户继续提交请求时, 负载均衡器会尝试向上次使用的地图服务器发送请求, 如果连接不上, 会尝试连接其他服务器, 从而为用户提供不间断的稳定的服务.

(3) 在地图服务器中, 缓存机制使得客户端与服务器保持连接状态, 当客户端由于某种原因丢失操作状态而在中途重新提交请求时, 地图服务器会从缓存池中获取用户操作的历史数据信息, 然后完成数据操作请求, 此过程对用户是透明的, 用户象之前正常情况一样获取相应的正确的处理结果. 这样, 使客户端可以持续访问地图服务器, 获得正确的空间数据处理结果.

4 性能分析

郭明强等(2008)提出了一种基于 socket 与地图服务器场的分布式 WebGIS 计算模型. 该模型虽然可以在一定程度上提高整个 WebGIS 系统的处理速度, 但是它是花费更多的硬件为代价, 其中的地图服务器的核心机制并未有所改善, 其缺点是当

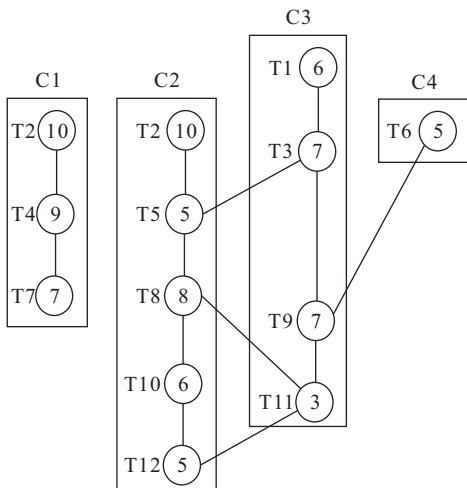


图 5 协作任务聚类

Fig. 5 The collaborative tasks cluster

表 2 硬件配置参数

Table 2 Hardware configuration parameter

角色	CPU 配置
地图服务器	CPU: AMD Athlon™64 X2 Dual Core Processor 4400+ 2.31 GHz; 2.00 GHz 内存: 2 G 硬盘: 464 G; 网卡: 100 M
	CPU: AMD Athlon™64 X2 Dual Core Processor 4400+ 2.31 GHz; 2.00 GHz 内存: 2 G 硬盘: 464 G; 网卡: 100 M
Web 服务器	CPU: Intel(R)Core™2 Duo 2.4 GHz; 2.39 GHz 内存: 2 G 硬盘: 250 G; 网卡: 100 M
负载均衡调度器	CPU: Intel(R)Pentium D 2.66 GHz; 2.68 GHz 内存: 2 G 硬盘: 185 G; 网卡: 100 M

并发访问用户数过大时,地图服务器不能正常地提供服务,导致很多请求得不到响应,甚至使服务器一直处于瘫痪状态.本文提出的 WebGIS 模型做了上述的改进,性能得到提高,在一定程度上减少了服务器大型空间计算的耗时,并提高了服务器在大用户量访问时的并发处理能力.

在 VS2005 环境下,对上文提到的基于服务+组件、基于 socket 与地图服务器场的分布式 WebGIS 计算模型以及本文提出的模型进行了性能对比.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.

通过 loadrunner 对系统进行压力负载测试.测试数据为 1 G 地图数据.图 6a 描述的是本模型与其他两个模型在加载地图事务时平均响应时间的比较.图 6b 描述的是本模型与其他两个模型在地图放大操作平均响应时间的比较.

从测试结果可以看出,原来的两种模型用户数越大,平均响应时间增长越快,当用户数超过 100 人时,地图加载事务平均响应时间已经超过 300 s,实际使用过程中用户很少会等待这么长的时间.而本文提出的基于聚类调度负载均衡的 WebGIS 模型在用户数量增加时,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.

5 总结与展望

本文针对目前 WebGIS 体系的弊端,提出基于聚类调度负载均衡的 WebGIS 模型,有效解决了 WebGIS 计算模型容错性差、并行处理能力弱、用户状态对象无法同步等问题,并对新旧 WebGIS 模型进行了性能对比测试,实验结果证实了本文提出的 WebGIS 模型在性能、容错性等方面有较高的提升;具有较好的稳定性、并发性、抗高负载能力和异常处理能力并且支持用户对服务器功能进行扩展以满足用户不断变化的需求.今后,下一步的研究工作将集中在如下两个方面:(1)地图服务器间的数据迁移;(2)大型空间计算的分解.

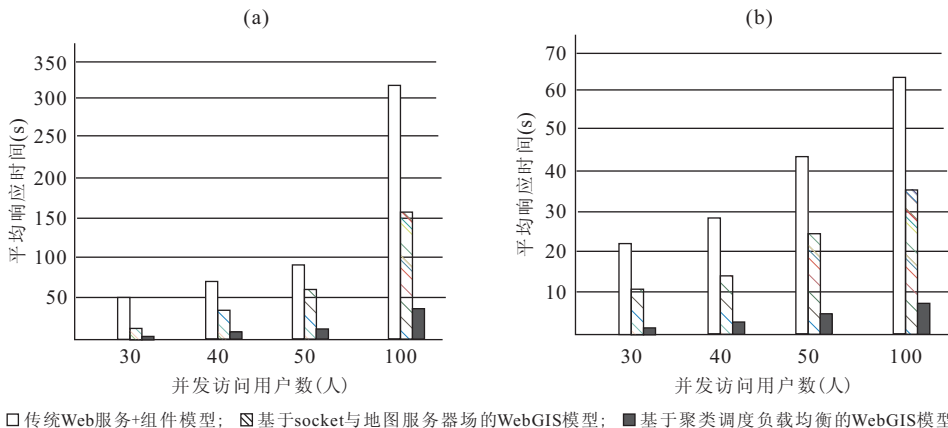


图 6 加载地图与地图放大事务平均响应时间比较

Fig. 6 The comparison of the average response time of load map
a. 加载地图事务平均响应时间; b. 地图放大操作平均响应时间

References

- Guo, M. Q., Huang, Y., Xie, Z., 2008. Design and implementation of distributed WebGIS model based on server farm. *Geography and Geo-Information Science*, 24(6): 12—17 (in Chinese with English abstract).
- He, J., Liu, R. Y., Liu, N., 2004. Technical design and systemic implementation of WebGIS based on COM+. *Journal of Zhejiang University (Science Edition)*, 31(6): 712—715, 720 (in Chinese with English abstract).
- Jiang, F., Zhou, B. Q., Wang, H. F., 2006. An effective load-balancing framework for distributed WebGIS. *Control & Automation*, 22(28): 215—217, 234 (in Chinese with English abstract).
- Li, G. H., Zhu, M. Z., Li, Y. M., 2008. Design and implementation of GIS application server. *Control & Automation*, 24(34): 203—205 (in Chinese with English abstract).
- Luo, X. G., Xie, Z., Wu, L., et al., 2006. Research and realization of WFS based on GML. *Earth Science—Journal of China University of Geosciences*, 31(5): 639—644 (in Chinese with English abstract).
- Tan, J., Yang, W. M., Zhang, B. P., et al., 2008. WebGIS based on self-defined protocol. *Computer Engineering*, 34(13): 248—250 (in Chinese with English abstract).
- Wang, H., Yu, Z. W., Li, R., et al., 2009. Cost based load balancing for network geographic information service. *Acta Geodaetica et Cartographica Sinica*, 38(3): 242—249 (in Chinese with English abstract).
- Wang, Y. E., Liu, H., 2009. Research on architecture and application of WebGIS. *Engineering of Surveying and Mapping*, 18(2): 70—73, 76 (in Chinese with English abstract).
- Wu, X. C., Wu, L., 2006. Service-oriented distributed spatial information supporting system. *Earth Science—Journal of China University of Geosciences*, 31(5): 585—589 (in Chinese with English abstract).
- Zhang, S., Kang, Z. W., 2006. Design and implementation of .net-based WebGIS system. *Computer Engineering*, 32(15): 106—108 (in Chinese with English abstract).
- Zhou, Z. B., Bian, F. L., 2006. Real-time monitoring system based on socket in WebGIS. *Science of Surveying and Mapping*, 31(4): 88—89, 69 (in Chinese with English abstract).

附中文参考文献

- 郭明强, 黄颖, 谢忠, 2008. 一种基于服务器场的分布式 WebGIS 计算模型设计与实现. *地理与地理信息科学*, 24(6): 12—17.
- 何江, 刘仁义, 刘南, 2004. 基于 COM+ 的 WebGIS 系统设计与实现. *浙江大学学报(理学版)*, 31(6): 712—715, 720.
- 江飞, 周保群, 王惠芳, 2006. 一种有效负载均衡的分布式 WebGIS 体系结构模型. *微计算机信息*, 22(28): 215—217, 234.
- 李光辉, 朱美正, 李艳明, 2008. GIS 应用服务器设计与实现. *微计算机信息*, 24(34): 203—205.
- 罗显刚, 谢忠, 吴亮, 等, 2006. 基于 GML 的 WFS 研究与实现. *地球科学——中国地质大学学报*, 31(5): 639—644.
- 谭靖, 杨为民, 张百平, 等, 2008. 基于自定义协议的网络地理信息系统. *计算机工程*, 34(13): 248—250.
- 王浩, 喻占武, 李锐, 等, 2009. 基于开销代价的网络地理信息服务负载均衡算法研究. *测绘学报*, 38(3): 242—249.
- 王玉娥, 刘晖, 2009. WebGIS 及其体系结构研究. *测绘工程*, 18(2): 70—73, 76.
- 吴信才, 吴亮, 2006. 面向服务的分布式空间信息支撑平台. *地球科学——中国地质大学学报*, 31(5): 585—589.
- 张胜, 康志伟, 2006. 基于 .net 技术的 WebGIS 系统的设计与实现. *计算机工程*, 32(15): 106—108.
- 周泽兵, 边馥苓, 2006. 基于 Socket 通信的 WebGIS 实时监测系统. *测绘科学*, 31(4): 88—89, 69.