

Web 端海量电网资源 GIS 数据动态调度与渲染方法^①

程 伟, 钱 蔚, 庞 宇, 杨文彬

(南瑞集团公司 (国网电力科学研究院), 南京 211006)

通讯作者: 程 伟, E-mail: chengwei5@sgepri.sgcc.com.cn

摘 要: 电网设备的图形展示是电力 GIS 应用的基础功能. 实现大数据量场景下 Web 端电网资源图的高性能显示已成为电力 GIS 系统建设需要解决的关键问题. 针对电网 GIS 数据及其应用特点, 本文提出了一种面向 Web 的海量电网资源 GIS 数据动态调度与渲染方法. 方法通过电网资源数据显示规则、线路数据分级压缩等方法控制前端数据加载量, 从而降低网络传输和前端绘制压力; 根据视口范围进行数据动态调度, 在客户端建立请求缓存、数据缓存以提高设备渲染效率; 使用 HTML5 技术进行数据的多线程解析和设备符号的实时绘制. 经验证, 在大数据量场景下, 本方法可显著提升现有电力 GIS 应用在 Web 图形显示方面的流畅性、交互性及灵活性, 大大改善用户体验.

关键词: 电力 GIS; 电网资源; 海量数据渲染; HTML5

引用格式: 程伟, 钱蔚, 庞宇, 杨文彬. Web 端海量电网资源 GIS 数据动态调度与渲染方法. 计算机系统应用, 2018, 27(2): 138-143. <http://www.c-s-a.org.cn/1003-3254/6228.html>

Method of Massive Power Resource Data Loading and Rendering for Web Client

CHENG Wei, QIAN Wei, PANG Yu, YANG Wen-Bin

(NARI Group Corporation (State Grid Electric Power Research Institute), Nanjing 211006, China)

Abstract: Display of power grid equipment on map is the basic function of power GIS application. According to the statistics, in most of our provinces, the amount of power grid resource data has reached tens of millions or even hundreds of millions. Therefore, the high performance display of massive power grid resource data on web has become a key problem to be solved in the construction of power GIS system. Aiming at the characteristics of GIS data and application of power grid, this paper presents a dynamic loading and rendering method for GIS data of massive power grid resources for Web. This method controls the amount of loaded data by data display rules and power line data compression, thereby reducing the network transmission and drawing pressure. It loads data according to the viewport bound, establishing request cache and data cache on the client to improve the efficiency of data rendering, and uses HTML5 technology for multi-threaded data analysis device symbols rendering. It is proved that this method can significantly improve the fluency, interactivity and flexibility of web client-side displaying for massive power resource GIS data, and has greatly improved the user experience.

Key words: power GIS; power grid resource; massive vector data rendering; HTML5

引言

电网发电、输电、变电、配电、用电等环节的业务应用均需要与地理空间信息集成融合, 以提升电网

资源管理水平和可视化分析能力^[1-8]. 电网设备的图形展示是电力 GIS 应用的基本功能. 据统计, 全国大部分省输变配低电网设备的数据量已达到数千万甚至上亿

^① 基金项目: 南瑞集团信息通信支撑智能电网重大专项 (524606160203)

收稿时间: 2017-05-17; 修改时间: 2017-06-05; 采用时间: 2017-06-22

级,因此,实现 Web 端海量电网 GIS 数据的高效显示已成为相关应用急需解决的问题。

目前,基于海量 GIS 数据的 Web 图形展示主要有以下几种实现方案:

(1) 实时动态栅格渲染. 该方案是目前 Web 端展示大规模电网 GIS 数据的主流方法. 主要思路是客户端根据当前视口范围向服务端发送图形渲染请求, 服务端根据请求范围从数据库中实时查询数据并渲染成图片返回至客户端显示. 该方法对客户端渲染能力要求不高, 其主要缺点为: 1) 服务端同时承担了数据检索和图形渲染的任务, 负载较大; 2) 由于 Web 端获取的是静态图片, 难以实现图形要素的闪烁、鼠标点击动态高亮等人机交互功能^[9]; 3) 客户端无法动态改变图形要素的颜色、样式, 导致诸如开关置位、设备带电着色、模拟停电分析等很多电力应用的重要需求难以满足.

(2) 栅格瓦片渲染. 该方法依据特定的二维空间分层网格切分方案, 在服务端预先生成一定空间范围和层数的栅格瓦片集, 即栅格瓦片金字塔. 客户端在地图浏览时根据视口范围从服务端获取对应的所有瓦片, 最终拼接成完整的地图. 由于是预先成图, 该方法降低了服务端实时绘图的压力, 但仍存在方案 (1) 的缺点 2) 和 3).

(3) 矢量瓦片渲染^[10-12]. 近几年, 随着 HTML5 技术的迅速发展, 大大提升了浏览器矢量绘图的能力. 由此发展出矢量瓦片技术, 该方法将矢量数据以建立金字塔的方式分割成一系列描述性文件, 以 GeoJson 或自定义格式组织. 客户端根据窗口范围请求对应的矢量瓦片数据进行绘图. 该方法克服了栅格渲染的缺点, 具有可交互、样式可控、体积小等优势. 但该方法会将空间对象分割存放在多个瓦片中, 损失了对象完整性, 极不利于业务功能的开发, 因此仍然只适用于地理底图的显示.

为解决现有电力 GIS 应用在 Web 端图形显示技术的不足, 本文充分利用 HTML5 的高性能矢量绘图能力及诸多新特性, 实现了一种面向 Web 的大规模电网 GIS 数据矢量化显示方法, 可显著提升现有电力行业 Web GIS 应用在图形显示方面的流畅性、交互性和灵活性, 改善用户体验.

1 设计

1.1 总体设计

本文方法中, 通过数据显示规则和线路 (含架空线和电缆) 抽稀, 合理控制图形浏览加载的数据量, 降低

网络传输及前端绘制压力; 在图形浏览时, 根据视口范围进行数据动态调度, 通过在前端建立请求缓存以减少请求次数, 建立数据缓存提升渲染速度; 最后使用 HTML5 技术进行 Web 端多线程数据解析与实时符号化渲染. 总体思路如图 1 所示.

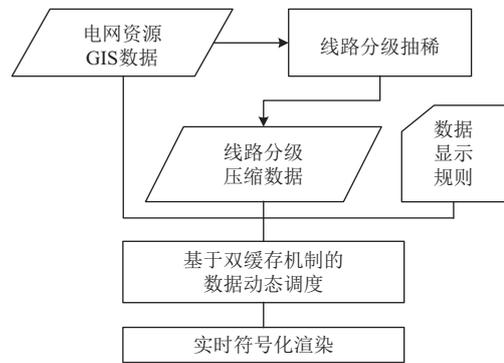


图1 方法总体设计图

1.2 数据显示规则制定

由于电网资源数据量大, 须设计一定的数据显示规则控制每屏显示的设备数量. 根据电力 GIS 应用的特点, 小比例尺地图上 (如 1:20 000 000) 只需显示高电压等级 (500KV 以上) 的电站和线路信息; 在大比例尺地图上 (如 1:1000), 需要显示电站的站内一次设备、杆塔及柱上设备. 基于以上需求特点, 本文设计了一种电网资源数据分级显示规则. 该规则首先确定地图最小显示比例尺和最大显示比例尺, 并将比例尺范围划分为若干连续区间, 每个区间称为一个显示级别, 如 L0, L1, L2, ..., Ln. 每个级别包含四个属性: 级别 ID、最小比例尺、最大比例尺、可见设备. 其中级别 ID 是显示级别的唯一标识; 最小比例尺和最大比例尺规定了该级别对应的比例尺范围; 可见设备指该显示级别下显示的设备类型及其电压等级.

客户端进行图形浏览时, 会根据当前地图比例尺确定显示级别, 再依据显示级别得到需要显示的设备类型及电压等级, 最后从服务端获取相应设备数据进行渲染.

1.3 线路数据压缩

输电线路是电力 GIS 应用展示的重要信息. 线路类型包括架空线和电缆线. 线路的数据量取决于节点个数, 而在电网数据模型中, 架空线的节点与杆塔一一对应, 由于杆塔数量巨大, 导致在显示大范围 (诸如全国、全省或全市范围) 电网线路时加载的线路数据量非常大, 这对数据传输、前端渲染都造成了严重负担.

因此,必须进行数据压缩以保证图形浏览的流畅性。

本文设计了一种线路数据分级压缩方法,主要思路是按照比例尺范围对线路进行分级抽稀,抽稀算法采用道格拉斯-普克算法^[13]。本方法中,制定若干压缩级别,每个压缩级别包含五个属性:级别ID、最小比例尺、最大比例尺、电压等级、抽稀阈值。其中级别ID是压缩级别的唯一标识;最小比例尺和最大比例尺规定了该级别对应的比例尺范围;电压等级限定了抽稀的电压等级范围(由于大范围的线路显示主要针对高压主干线路,因此具有较高电压等级的线路才需要压缩);抽稀阈值指该级别采用的道格拉斯阈值,阈值越大,抽稀力度越大。抽稀后的线路数据可缓存于查询服务进行的内存或存储在空间数据库中。

2 实现

2.1 数据动态调度

本文中,数据加载采用按需调度和本地缓存的策略。

按需调度的基本原理是:虽然数据总量很大,但每屏显示的数据量总是有限,因此可以根据显示需要只加载较少的数据。具体包括两方面:根据当前视口范围确定数据调取的空间范围;根据当前地图比例尺,结合数据显示规则确定需要调取的设备类型和电压等级。通过以上两层条件的过滤,可以大大降低数据加载量。

为提高显示效率,本文在客户端建立请求缓存和数据缓存。请求缓存存储了客户端每次发送的数据查询请求,内容为请求的空间范围和显示级别。其作用是快速判断一个新数据请求是否被已有请求包含,如已被包含,则客户端无需发送,将减少重复请求。数据缓存存储了每次查询返回的设备信息,包含空间位置信息和属性信息,用于缓存所有已获取的设备数据。本文中,数据缓存作为客户端设备绘制的直接数据源,即客户端会从请求缓存中快速检索出当前视口范围的设备对象进行图形绘制。

由于在两个缓存的使用过程中都涉及按空间范围查找已有请求或设备数据。因此,在缓存的存储结构上采用了R树^[14]结构。R树是目前广泛采用的一种空间索引数据结构。通过R树,可快速检索出指定空间范围内的空间对象。它是一个高度平衡树,由中间节点和叶节点组成。每个结点都对应一个区域,中间节点存储其所有子节点的区域范围,其所有子节点的区域都落在它的区域范围之内;叶节点的存储其区域范围之内的所有空间对象的外接矩形。根据R树的这种数据结构,

当需要查询指定空间范围内的对象时,只需要遍历少数几个叶节点所包含的空间对象,判断是否满足条件即可。这种方式使不必遍历所有对象,效率显著提高。

数据动态调度的主要过程如下:

(1) 客户端缓存容器创建。

客户端建立两个R树结构作为请求缓存和数据缓存,分别记为 $RTreeRequest$ 、 $RTreeData$ 。每个数据请求作为 $RTreeRequest$ 中存储的一个条目,记为 (I, L) 。 I 为条目最小外接矩形,是数据请求的空间矩形范围, L 为数据请求对应的显示级别ID。在 $RTreeData$ 中每个设备作为一个存储条目,记为 $(R, Data)$ 。 R 为条目的外接矩形,是设备空间图形的最小外接矩形, $Data$ 是设备的空间信息和属性信息。

(2) 客户端确定显示级别及空间范围。

获取当前地图比例尺,根据比例尺在数据显示规则中匹配对应的显示级别,记为 LC 。客户端获取当前屏幕显示的地理空间范围(为一个矩形范围),对该范围进行适当外扩,如范围的长、宽分别外扩20%,形成数据查询空间范围,记为 IC 。进行外扩的原因是客户端将缓存本次请求的空间范围,当进行地图平移浏览时,新的屏幕空间范围如过没有超出上次数据请求的实际空间范围,则无需再次请求。这样可降低数据请求次数,减轻服务端压力。

(3) 客户端发送数据查询请求。

首先利用R树检索算法从 $RTreeRequest$ 中快速查找所有与 IC 有空间重叠的请求条目,形成集合 A 。遍历 A 中所有请求条目,如存在某个请求条目 (In, Ln) ,其查询范围 In 包含 IC 且显示级别 Ln 等于 LC ,则无需发送请求,流程终止;如没有符合上述条件的请求条目,则向服务端发送数据查询请求,请求参数包括空间矩形范围 IC 、显示级别 LC 及当前比例尺 SC 。发送完成后,将本次请求条目 (IC, LC) 插入至 $RTreeRequest$ 中以缓存。

(4) 服务端执行查询并返回结果。

服务端根据请求参数确定查询条件。其中空间范围条件为 IC 。属性条件为设备类型和电压等级过滤条件,具体由显示级别 LC 结合显示规则确定。

对非线性设备,服务端根据上述空间条件、属性条件从空间数据库中查询数据并以GeoJson格式返回给客户端。对线路设备,首先根据本次请求的比例尺 SC 、电压等级条件由数据压缩规则确定查询数据源。如需查询原始数据,处理过程与非线性设备相同;如需

查询压缩数据,则计算压缩级别,从服务端内存缓存或空间数据库中查询对应级别的压缩数据并返回给客户端.

(5) 客户端进行设备数据解析与缓存.

客户端从服务端获取查询结果后,需将 GeoJson 字符串解析为内存对象并保存在本地数据缓存 *RTreeData* 中.经实测,当返回的数据量较大时,数据解析过程较为耗时,此时页面的状态是不可响应的,导致图形浏览出现卡顿现象.为了提高图形浏览的流畅性,本文采用

HTML5 提供的多线程解决方案进行数据解析和重复数据剔除.具体方法为:客户端预先开辟一个 Web Worker 后台线程,页面主线程获得查询结果 GeoJson 字符串后,将字符串和已加载设备 ID 集合传给后台线程,后台线程执行数据解析生成内存对象,并根据已加载设备 ID 集合剔除已加载设备,形成新加载设备对象的集合传给主线程.主线程获取新加载设备对象集合后,将其全部插入本地设备数据缓存 *RTreeData* 中.

数据调度的详细流程如图 2 所示.

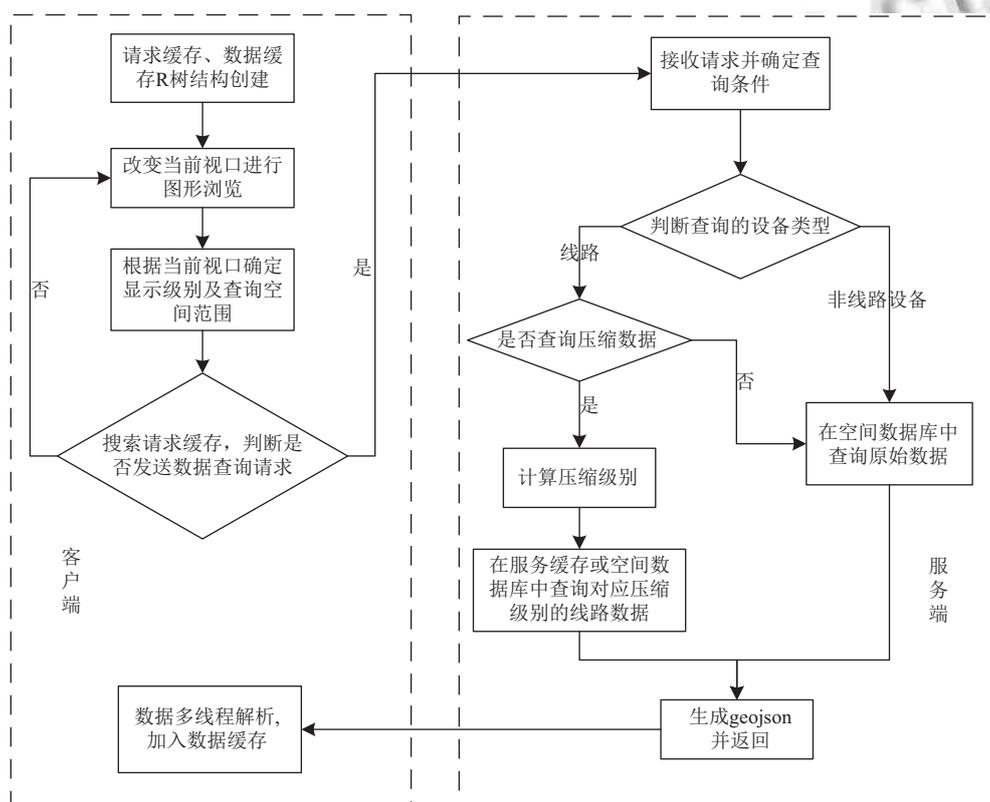


图 2 数据动态调度流程

2.2 数据实时渲染

电力 GIS 应用有很多重要需求,诸如设备高亮显示、模拟开关置位、模拟带电着色、模拟停电分析等都需要在客户端实时改变设备的颜色、符号、样式.目前常规的服务端渲染方式难以满足上述需求,而本文采用客户端实时渲染策略,最大的优势是可以动态改变设备图形的颜色、样式,可以很方便的实现上述功能.

当用户进行地图浏览使地图视口发生变化时,客户端将开始数据实时渲染.其过程主要分为两步:第一步,基于 R 树搜索方法从本地数据缓存中快速抽取当前视口地理范围内的设备对象,形成待渲染设备对象

集合.第二步,对集合中的每个设备,获取该设备的符号信息,并进行符号的绘制.

本文采用 HTML5 提供的 Canvas 元素作为地图绘制的画布,在画布 (Canvas) 上通过 JavaScript 脚本对基础地理及电网设备符号进行实时绘制.

电网设备图形符号主要分为两类:线状符号和点状符号.线状符号指电缆、架空线、站内连接线等设备的符号,用实线或虚线表示,其绘制比较简单.点状符号包括变压器、电流互感器、断路器等绝大部分电网设备(多达几百种)符号.每种点状符号由若干基本图元(折线、多边形、圆、椭圆、椭圆弧、贝塞尔曲线、文字等)组合而成.本文针对每个点符号都新建一

个 Canvas 元素, 利用 Canvas 提供的图形绘制接口完成符号所有基本图元的绘制, 然后将其绘制到地图的 Canvas 上, 从而实现设备符号的实时绘制. 与常规的使用静态图片 (如 png、svg) 显示设备符号的方式相比, 这种实时绘制的方法可以动态更改符号的颜色, 很好的满足了电网运行分析应用所需要的动态着色需求.

客户端实现设备符号化渲染的实现流程如图 3 所示, 主要包括如下步骤.

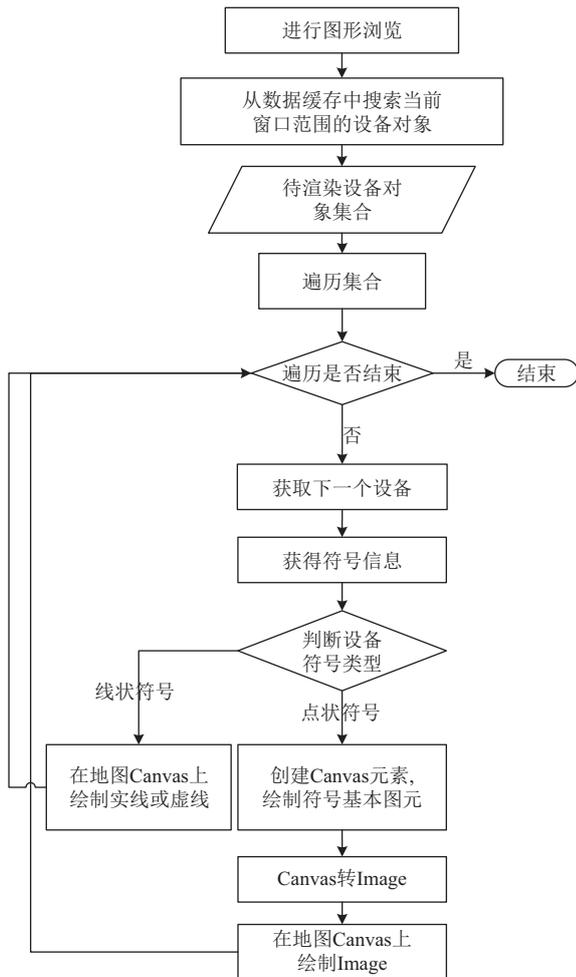


图 3 数据实时渲染流程

- (1) 开始图形浏览, 进行地图平移或缩放, 改变当前视口范围;
- (2) 从数据缓存中搜索当前视窗范围的设备对象, 形成待渲染设备集合, 遍历待渲染设备集合;
- (3) 判断遍历是否结束, 如果是则结束, 如果否则转向步骤 (4);
- (4) 获取下一个设备, 获取设备对应的符号, 判断设备符号类型, 如果为线状符号, 直接在地图的画布 (Canvas) 上绘制实线或虚线, 并转向步骤 (3); 如果为点

状符号, 首先新建一个画布 (Canvas) 元素, 在该元素上绘制符号的基本图元; 然后将画布 (Canvas) 元素转为图片 (Image) 元素; 最后使用 HTML5 提供的 drawImage 方法将图片绘制于地图的画布上, 并转向步骤 (3).

3 测试与验证

3.1 测试数据与程序实现

基于某省电网资源数据进行电网资源图形展示测试. 该省电网设备 (含低压设备) 数量总计 1.4 亿以上, 其中线路 (包含架空线段和电缆线段) 数据量在 1500 万以上.

测试的电网资源空间数据库采用 MongoDB 3.2, 查询服务采用 Java 实现, Web 端采用 JavaScript 与 HTML5 实现.

3.2 图形渲染效果

图 4 为较大地理范围下的大型电站和重要线路的显示效果; 图 5 为地图放大后站内接线图的渲染效果.

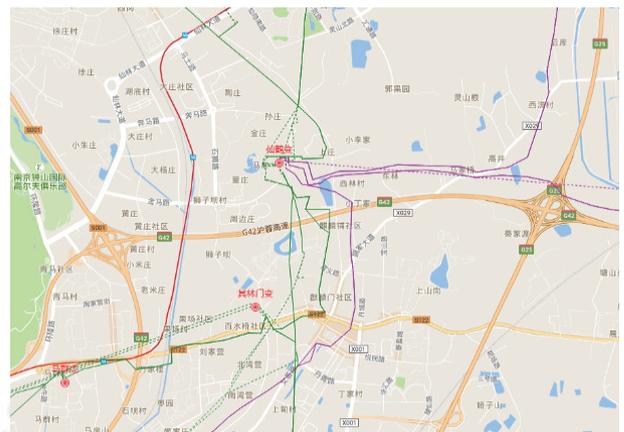


图 4 电网资源图渲染效果——电站和线路分布

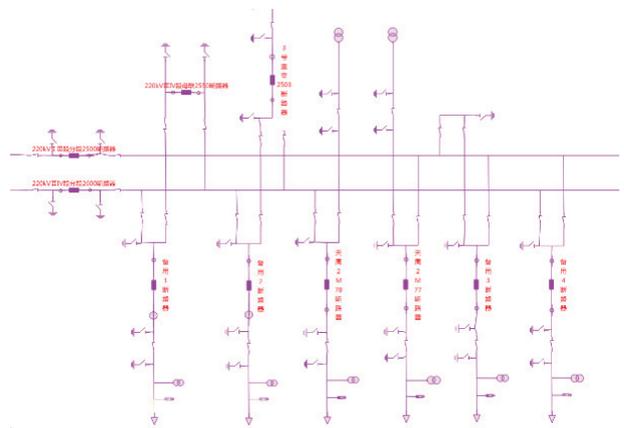


图 5 电网资源图渲染效果——站内接线图

3.3 缓存与数据压缩效果

在客户端进行5次独立的随机浏览测试,每次浏览均包含地图缩小、放大、平移操作.测试记录了浏览过程中客户端视口变化次数、实际线路查询请求次数.从测试结果看,由于利用请求缓存避免了一些重复请求,使数据请求次数明显少于视口变化次数,可见请求缓存在降低服务端数据查询、客户端数据解析压力方面效果明显.另外,为验证线路数据压缩效果,线路查询请求同时返回原始数据及压缩后的数据,统计两种线路数据的总量.从结果对比看,通过线路压缩,大大降低了数据传输量,进而减轻了客户端渲染压力.具体测试结果见表1.

表1 客户端随机漫游测试结果

测试编号	视口变化次数	请求次数	接收线路总数据量	
			未压缩	压缩
测试1	35	13	6.7	3.2
测试2	32	20	6.2	3.8
测试3	47	32	14.9	6.2
测试4	27	15	7.9	4.5
测试5	34	18	20.6	8.6

3.4 比较分析

目前,国家电网公司电力GIS应用主要集中在生产管理、配网抢修等业务系统.上述系统中Web端电网图形展示方法为:每次地图视口变化时,客户端向服务端发送图形请求,服务端即时从数据库中查询数据并渲染成图片传给客户端显示,显示组件采用Flex.与以上方案相比,本文方法主要有以下两点优势:

(1) 本文方法采用客户端实时渲染,比服务端渲染方式交互性更好,且可以即时改变设备显示的颜色、样式.因此,本文方法更易于实现设备高亮、设备点击查询、模拟开关置位、带电着色等常用功能.此外,图形渲染由客户端完成,网络传输数据量小、服务器压力轻.

(2) 上述系统中每次视口变化都进行服务渲染请求,而本文方法由于使用了请求缓存减少了重复请求,避免了每次视口变化都发送数据请求,降低了服务端压力.

4 结论

本文针对电力GIS数据及应用特点,提出一种面向Web的海量电网资源GIS数据动态调度及矢量化显示方法.经实测验证:该方法可支持海量电网GIS数

据在Web端的高性能显示;由于采用Web端矢量化渲染,非常适合电网运行分析应用所需要的模拟开关置位、设备动态带电着色和动态标注数据刷新的需求,同时便于实现对象闪烁、鼠标移动/点击动态高亮等效果,能够提供更好的用户体验.本文研究成果已应用于多个大型电网GIS应用系统.

参考文献

- 史兴华. 电网GIS及其应用. 北京: 中国电力出版社, 2010.
- 张德金, 吴红卫, 余运波. 基于统一电网资源模型的电网资源数据维护机制和关键技术的研究. 电气应用, 2015, (S2): 423-427.
- 唐凯, 钟一俊, 吴惠芬, 等. 电网GIS基础数据质量管理工具设计与应用. 电力信息与通信技术, 2016, 14(2): 98-101.
- 易勇强, 奚建飞, 雷伟刚. 电网GIS多源多时相多分辨率空间数据质量管控技术体系研究. 南方能源建设, 2016, 3(S1): 122-125.
- 王瀚秋, 刘开锋. 低压配电网GIS结合电力营配一体化信息平台的建设与应用. 电力信息化, 2010, 8(1): 71-74.
- 刘才华. 电网GIS地理信息服务平台建设与应用效益分析. 数字技术与应用, 2012, (10): 235-236.
- 林峰, 郭宝, 钱蔚. 面向公共电网GIS平台的电网地理图形应用架构. 电力系统自动化, 2011, 35(24): 63-67.
- 陈健, 万国成, 孙冠男, 等. GIS平台电网规划辅助决策系统开发与应用. 电力系统及其自动化学报, 2010, 22(6): 99-103.
- Gaffuri J. Toward web mapping with vector data. Proceedings of the 7th International Conference on Geographic Information Science. Columbus, OH, USA. 2012. 87-101.
- Mapbox Co. Mapbox vector tile specification. <https://www.mapbox.com/vector-tiles/specification/>. [2016-01-19].
- 朱秀丽, 周治武, 李静, 等. 网络矢量地图瓦片技术研究. 测绘通报, 2016, (11): 106-109, 117.
- 孙晨龙, 霍亮, 高泽辉. 基于矢量瓦片的矢量数据组织方法研究. 测绘与空间地理信息, 2016, 39(4): 122-124.
- Cromley RG, Campbell GM. Integrating quantitative and qualitative aspects of digital line simplification. The Cartographic Journal, 1992, 29(1): 25-30. [doi: 10.1179/caj.1992.29.1.25]
- Guttman A. R-trees: A dynamic index structure for spatial searching. Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data. Boston, MA, USA. 1984. 47-54.