



(12)发明专利申请

(10)申请公布号 CN 106445809 A

(43)申请公布日 2017.02.22

(21)申请号 201610766138.1

(22)申请日 2016.08.30

(71)申请人 北京华恒铭圣科技发展有限责任公司

地址 100016 北京市朝阳区酒仙桥路甲10号星城国际A座501室

(72)发明人 李万林 张洋 李克

(74)专利代理机构 北京思海天达知识产权代理有限公司 11203

代理人 沈波

(51)Int.Cl.

G06F 11/36(2006.01)

G06F 17/30(2006.01)

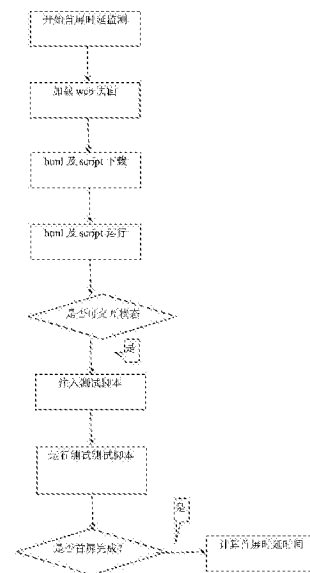
权利要求书2页 说明书8页 附图2页

(54)发明名称

一种获取网页加载首屏时延方法

(57)摘要

一种获取网页加载首屏时延方法,属于网络和业务测试领域。首先获得并预置目标网站的布局;资源在html已经下载完成,且document.readyState已经在处于interactive状态同时,在目标网站的网页加载资源开始时注入js代码,轮询检查计算当前网页中加载的资源元素是否在首屏范围内及是否已经加载显示。一旦在首屏的资源符合布局并加载完成,则计算从用户请求网页开始到当前的时延,记为首屏时延。与现有技术相比较,本发明操作简单,用户只需点击功能按钮便可进行测试,无需其他繁琐操作;耗电低,本方法仅仅简单的脚本程序运行,占用cpu及内存资源极低,不使用其他的耗电硬件,对用户无感觉。计算速度快、安全可靠、普及性高、适用性广,易于大范围推广。



1. 一种获取网页加载首屏时延方法,其特征在于:网页首屏所需要加载的资源已经在html页面上加载完成;

由于首屏时延所包含的主要部分在于资源的下载和加载,所以本方法通过在webview中注入javascript结合网站布局动态检查屏幕状态来确定首屏时延,具体步骤如下:

步骤1) 首先获得并预置目标网站页面的布局;

步骤2) 资源在html已经下载完成,且document.readyState已经在处于interactive状态同时,在目标网站的网页加载资源开始时注入js代码,轮询检查计算当前网页中加载的资源元素是否在首屏范围内及是否已经加载显示;

步骤3) 一旦在首屏的资源符合布局并加载完成,则计算从用户请求网页开始到当前的时延,记为首屏时延;

所述资源是指网页展示所需要加载的全部内容,包括文本、图片、脚本、flash。

2. 根据权利要求1所述的一种获取网页加载首屏时延方法,其特征在于:实现本方法的架构包括操作系统、应用程序框架层、Webview-FrameWork层、应用程序、运行脚本;

OS-Android操作系统,包含Linux2.6内核及驱动,对系统安全性、内存、进程、网络协议栈及硬件层进行管理;

应用程序框架层包括活动管理器、窗口管理器、内容提供者、视图系统、包管理器、电话管理器、资源管理器、位置管理器、通知管理器和XMPP服务等部分;应用程序框架层为FrameWork,FrameWork是Android应用程序的运行基础,以提供android应用程序所使用的API框架;

Webview-FrameWork层提供公共控件,Android应用程序能够使用webview进行网页加载并运行javascript等操作;

应用程序为指APP-Android首屏时延监测程序;

运行脚本为JavaScript-java脚本,能够在html加载时运行,由webview提供运行环境。

3. 根据权利要求1所述的一种获取网页加载首屏时延方法,其特征在于:架构总体描述如下,

首屏时延监测程序运行在APP层,使用webview加载网页,并在html加载过程中注入javascript脚本获得首屏时延时序图步骤描述前置条件:

预先获得目标网站的页面布局特点;

步骤1:首屏时延监测程序使用webview加载目标网站;

步骤2:webview首先下载html并进行加载,在加载过程中会以此触发loading、interactive、complete几个状态,当加载到interactive时表示网页已经开始响应与用户之间的交互,譬如交互的方式为点击操作,但仍可能有一部分资源在加载;

步骤3:在interactive状态时注入javascript;

步骤4:javascript开始运行,首先根据当前网站的布局特点,遍历所有html已加载的资源元素及根据资源位置计算是否属于首屏,同时也确定该资源是否已加载显示;如果已经加载显示的资源全部符合布局要求,则判定为首屏加载完毕,考虑的加载的滞后性,需要重复执行几次;

步骤5:首屏判定完毕返回给主程序,从加载目标网站开始到此计算为首屏时延。

4. 根据权利要求1所述的一种获取网页加载首屏时延方法,其特征在于:流程图步骤描

述:

步骤1:开始执行首屏时延监测任务;

步骤2:使用webview加载目标网站;

步骤3:首先下载html及script;

步骤4:在html下载完毕同时运行html及相关脚本;

步骤5:在加载过程中会以此触发loading、interactive、complete几个状态,当加载到interactive时表示网页已经响应用户交互(譬如点击操作),但仍可能有一部分资源在加载;

步骤6:在interactive状态时注入测试javascript;

步骤7:javascript开始运行,首先根据当前网站的布局特点,遍历所有html已加载的资源元素及根据资源位置计算是否属于首屏,同时也确定该资源是否已加载显示;如果已经加载显示的资源全部符合布局要求,则判定为首屏加载完毕,考虑加载的滞后性,需要重复执行几次;

步骤8:首屏判定完毕返回给主程序,从加载目标网站开始到此计算为首屏时延。

## 一种获取网页加载首屏时延方法

### 技术领域

[0001] 本发明涉及网络优化测试领域,更具体地,本发明涉及获取网页加载首屏时延的方法。

### 背景技术

[0002] 在现有的网络状态下,使用者经常会遇到带宽拥塞、应用性能低下、蠕虫病毒、DoS肆虐、恶意入侵等对网络使用及资源有负面影响的问题及困扰,网络优化功能是针对现有的防火墙、安防及入侵检测、负载均衡、频宽管理、网络防毒等设备及网络问题的补充,能够通过接入硬件及软件操作的方式进行参数采集、数据分析,找出影响网络质量的原因,通过技术手段或增加相应的硬件设备及调整使网络达到最佳运行状态的方法,使网络资源获得最佳效益,同时了解网络的增长趋势并提供更好的解决方案。

[0003] 实现网络应用性能加速、安全内容管理、安全事件管理、用户管理、网络资源管理与优化、桌面系统管理,流量模式监控、测量、追踪、分析和管理,并提高在广域网上应用传输的性能的功能的产品。主要包括网络资源管理器,应用性能加速器,网页性能加速器三大类,针对不同的需求及功能要求进行网络优化。

[0004] 现有的运营商网络优化模式中,首先需要有效获得网页首屏加载时延以衡量主流网站用户感知的流畅度。随着移动网络带宽的提高以及手机用户数据使用的不断普及,运营商需要以用户的视角来衡量网络及web网页的流畅度,而web网页里最直观最重要地就是首屏显示时延(即用户看到并可操作的时延),而该时延直接反映了用户对一般网站流畅度的主观认知,一般使用普通终端抓包或网络侧无法直接获得,从而使该指标无法简单有效量化并分析。

[0005] 目前存在的首屏时延方式是通过截屏来比对分析图像,这种方法限制较大,一旦网站内容更换也需要更换样本图片才能确保正常工作;且截图方式在手机端操作步骤复杂、耗电高、精确性低,因此,不适合大规模推广。

### 发明内容

[0006] 本发明的目的在于协助运营商直观了解网站用户的体验感知,以达到简单方便获得网站交互性感知时延,为此本发明提出了一种获取网页首屏时延的方法。

[0007] 为实现上述目的,本发明采用的技术方案为一种获取网页加载首屏时延方法,其特征在于:网页首屏所需要加载的资源已经在html页面上加载完成;

[0008] 由于首屏时延所包含的主要部分在于资源的下载和加载,所以本方法通过在webview中注入javascript结合网站布局动态检查屏幕状态来确定首屏时延,具体步骤如下:

[0009] 步骤1) 首先获得并预置目标网站页面的布局;

[0010] 步骤2) 资源在html已经下载完成,且document.readyState已经在处于interactive状态同时,在目标网站的网页加载资源开始时注入js代码,轮询检查计算当前

网页中加载的资源元素是否在首屏范围内及是否已经加载显示;

[0011] 步骤3)一旦在首屏的资源符合布局并加载完成,则计算从用户请求网页开始到当前的时延,记为首屏时延;

[0012] 所述资源是指网页展示所需要加载的全部内容,包括文本、图片、脚本、flash。

[0013] 与现有技术相比较,本发明具有如下有益效果。

[0014] 1、操作简单,用户只需点击功能按钮便可进行测试,无需其他繁琐操作。

[0015] 2、耗电低,本方法仅仅简单的脚本程序运行,占用cpu及内存资源极低,不使用其他的耗电硬件,对用户无感觉。

[0016] 3、计算速度快,用户在看到网页加载完毕时即可得知首页加载时延,不需要额外等待。

[0017] 4、安全可靠,代码运行在专门测试程序里,不对普通用户产生任何影响。

[0018] 5、普及性高,普通android2.2以上手机均可运行,适用性广,易于大范围推广。

[0019] 总体而言,相对于截图对比的方式,本设计操作简单、耗电低、计算速度快,安全可靠,适合大范围推广

#### 附图说明

[0020] 图1是根据本发明进行的总体结构图。

[0021] 图2是本发明的工作时序图。

[0022] 图3是本发明的流程图。

#### 具体实施方式

[0023] 以下结合附图1-3对本发明作进一步详细说明。

[0024] 如图1所示,实现本发明方法的架构包括操作系统、应用程序框架层、WebView-FrameWork层、应用程序、运行脚本。

[0025] OS-Android操作系统,包含Linux2.6内核及驱动,对系统安全性、内存、进程、网络协议栈及硬件层进行管理。

[0026] 应用程序框架层包括活动管理器、窗口管理器、内容提供者、视图系统、包管理器、电话管理器、资源管理器、位置管理器、通知管理器和XMPP服务等部分。应用程序框架层为FrameWork,FrameWork是Android应用程序的运行基础,以提供android应用程序所使用的API框架。

[0027] WebView-FrameWork层提供公共控件,Android应用程序能够使用webview进行网页加载并运行javascript等操作。

[0028] 应用程序为指APP-Android首屏时延监测程序。

[0029] 运行脚本为JavaScript-java脚本,能够在html加载时运行,由webview提供运行环境

[0030] 架构总体描述

[0031] 首屏时延监测程序运行在APP层,使用webview加载网页,并在html加载过程中注入javascript脚本获得首屏时延时序图步骤描述前置条件:

[0032] 预先获得目标网站的页面布局特点,目标网站需要加载两张360\*180,3张90\*67的

图片。

[0033] 步骤1:首屏时延监测程序使用webview加载目标网站,譬如新浪网、搜狐网、凤凰网等。

[0034] 步骤2:webview首先下载html并进行加载,在加载过程中会以此触发loading、interactive、complete几个状态,当加载到interactive时表示网页已经开始响应与用户之间的交互,譬如交互的方式为点击操作,但仍可能有一部分资源在加载。

[0035] 步骤3:在interactive状态时注入javascript。

[0036] 步骤4:javascript开始运行,首先根据当前网站的布局特点,遍历所有html已加载的资源元素及根据资源位置计算是否属于首屏,同时也确定该资源是否已加载显示;如果已经加载显示的资源全部符合布局要求,则判定为首屏加载完毕,考虑的加载的滞后性,需要重复执行几次;

[0037] 步骤5:首屏判定完毕返回给主程序,从加载目标网站开始到此计算为首屏时延。

[0038] 流程图步骤描述:

[0039] 步骤1:开始执行首屏时延监测任务;

[0040] 步骤2:使用webview加载目标网站;

[0041] 步骤3:首先下载html及script;

[0042] 步骤4:在html下载完毕同时运行html及相关脚本;

[0043] 步骤5:在加载过程中会以此触发loading、interactive、complete几个状态,当加载到interactive时表示网页已经响应用户交互(譬如点击操作),但仍可能有一部分资源在加载;

[0044] 步骤6:在interactive状态时注入测试javascript;

[0045] 步骤7:javascript开始运行,首先根据当前网站的布局特点,遍历所有html已加载的资源元素及根据资源位置计算是否属于首屏,同时也确定该资源是否已加载显示;如果已经加载显示的资源全部符合布局要求,则判定为首屏加载完毕,考虑加载的滞后性,需要重复执行几次;

[0046] 步骤8:首屏判定完毕返回给主程序,从加载目标网站开始到此计算为首屏时延;

[0047] 主要JavaScript代码

```
if(curUrl.indexOf("sina.")>-1){
```

```
    var template=new Object();
```

```
    template.width =360;
```

[0048] template.height=180;

```
    template.sum=0;
```

```
    template.defSum=2;
```

```
    templatesArry.push(template);
```

```
var template=new Object();
template.width =90;
template.height=67;
template.sum=0;
template.defSum=3;
templatesArray.push(template);
}

document.addEventListener('DOMContentLoaded', function() {

    window.js_method.notify("DOMContentLoaded");
    tickEnabled=true;

}, false);
```

[0049]

```
document.onreadystatechange = function() {

    if (document.readyState == "interactive") {

        window.js_method.notify("interactive");

    } else if (document.readyState == "complete") {

        if (timer) {
            clearInterval(timer);
        }

        if(!allImgLoaded){
            allImgLoaded=true;
            window.js_method.firstScreenFin();
        }
    }
}
```

```
    }  
  
    window.js_method.notify("complete");  
  }  
};  
  
var timer = setInterval(function() {  
  
    if(!tickEnabled){  
        return;  
    }  
  
    var obj = document.querySelectorAll('IMG');  
  
    for (var i = 0; i < obj.length; i++) {  
[0050]         var elLeft=obj[i].getBoundingClientRect().left +  
document.documentElement.scrollLeft;  
            var elPos = obj[i].getBoundingClientRect().top +  
document.documentElement.scrollTop;  
            var elHeight = obj[i].scrollHeight;  
            var elWidth=obj[i].scrollWidth;  
  
            if (obj[i].src !== null &&  
                obj[i].src !== undefined &&  
                obj[i].src !== "" &&  
                elHeight>0 && elPos>0 && elWidth>0 &&  
                (elPos + elHeight) <= screenLimitHeight ) {  
  
                var picExist=false;
```



```
if (firstScreenImgs.length > 0) {  
  
    for (var j = 0; j < firstScreenImgs.length; j++) {  
  
        var img = firstScreenImgs[j];  
  
        if (img.src == obj[i].src) {  
  
            picExist=true;  
  
            break;  
        }  
  
    }  
}  
[0051] }  
  
if(!picExist){  
  
    firstScreenImgs.push(obj[i]);  
}  
}  
}  
  
for (var i = 0; i < firstScreenImgs.length; i++) {  
  
    var img = firstScreenImgs[i];  
  
    if (img.complete) {  
  
        for(var j=0;j<templatesArry.length;j++){
```

```
        if(templatesArray[j].width==img.scrollWidth &&
            Math.abs(templatesArray[j].height-img.scrollHeight)<5
        &&
            templatesArray[j].sum<templatesArray[j].defSum){
            templatesArray[j].sum++;
        }
    }
}

        var elLeft=img.getBoundingClientRect().left +
document.documentElement.scrollLeft;
        var elPos = img.getBoundingClientRect().top +
document.documentElement.scrollTop;
        var elHeight = img.scrollHeight;
[0052]    var elWidth=img.scrollWidth;
    }

var templateOK=true;

for(var j=0;j<templatesArray.length;j++){

    if(templatesArray[j].sum<templatesArray[j].defSum){
        templateOK=false;
    }
}

if (templateOK && !allImgLoaded){
    allImgLoaded=true;
    window.js_method.firstScreenFin();
}
```

}

[0053]

}, 1000)。

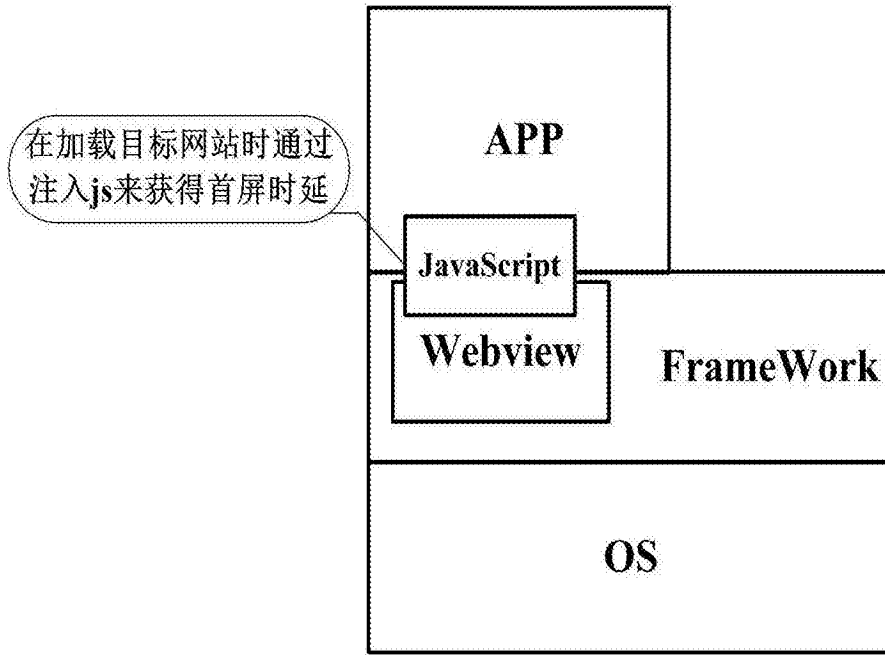


图1

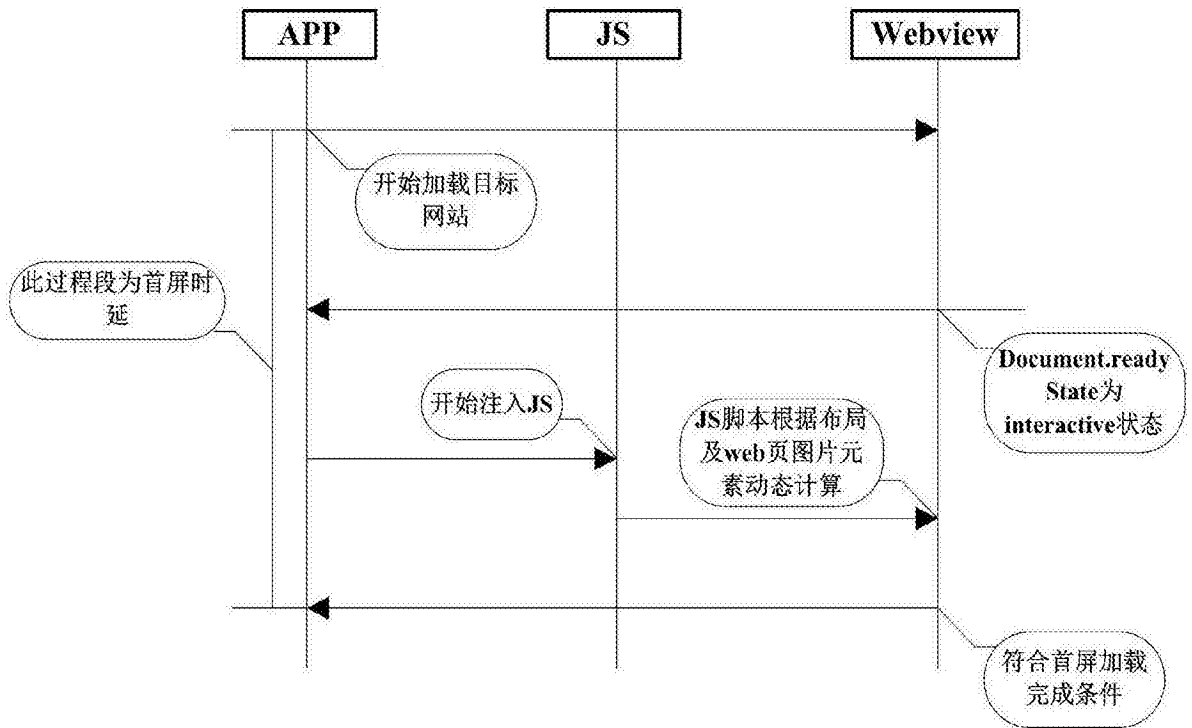


图2

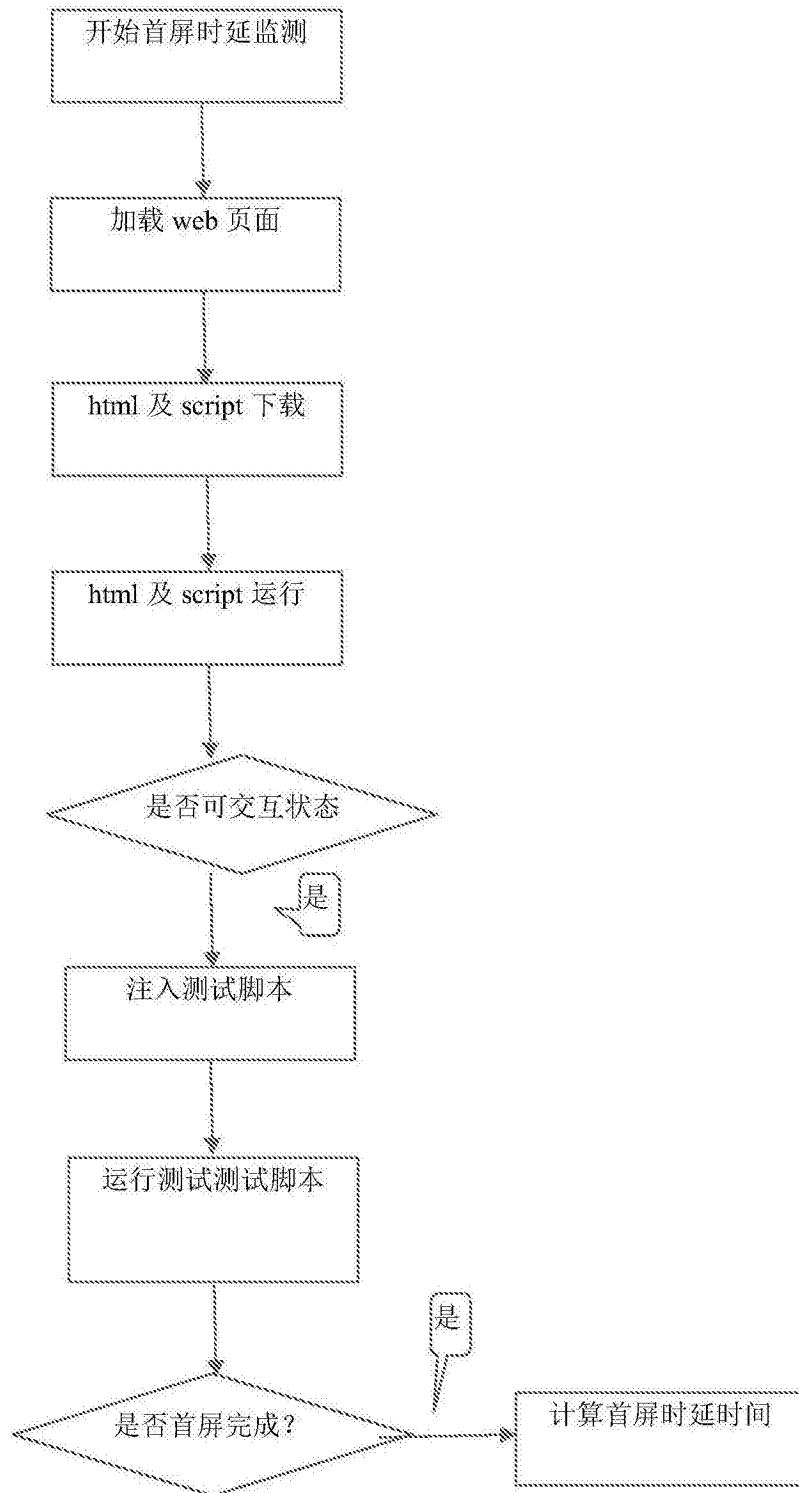


图3