



(12)发明专利

(10)授权公告号 CN 104010031 B

(45)授权公告日 2018.01.05

(21)申请号 201410215420.1

(22)申请日 2014.05.21

(65)同一申请的已公布的文献号

申请公布号 CN 104010031 A

(43)申请公布日 2014.08.27

(73)专利权人 华南理工大学

地址 511400 广东省广州市南沙区环市大道南路25号华工大广州产研院

(72)发明人 李志瑞 覃健诚 陆以勤

(74)专利代理机构 广州粤高专利商标代理有限公司 44102

代理人 何淑珍

(51)Int.Cl.

H04L 29/08(2006.01)

(56)对比文件

CN 103258015 A, 2013.08.21,

CN 101141449 A, 2008.03.12,

CN 101911603 A, 2010.12.08,

CN 101960449 A, 2011.01.26,

穆林涛.“基于Smart Client和Web服务的诊疗系统的实现”.《中国优秀硕士学位论文全文数据库 信息科技辑》.2006,

刘近勇 等.“Ajax技术与Web应用”.《邮电设计技术》.2007,

审查员 李福涛

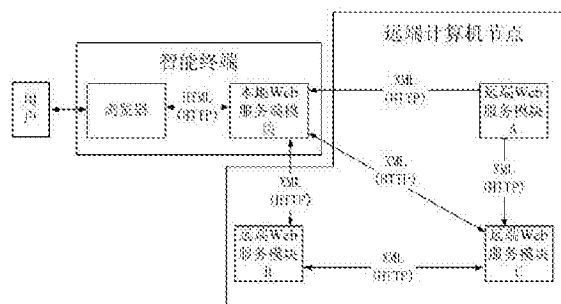
权利要求书2页 说明书5页 附图3页

(54)发明名称

一种改善Web浏览器应用效率的分布式协同方法

(57)摘要

本发明公开一种改善Web浏览器应用效率的分布式协同方法。该方法通过运行于智能终端本地的Web服务端模块来实现强于一般浏览器端的功能。本地Web服务端模块与包括云端服务器在内的远端Web服务端模块进行交互成为分布式协同计算的一个节点。浏览器通过与该本地Web服务端模块进行交互获取网页文件进行显示。浏览器与该本地Web服务端模块的交互不受网络状况影响,能显著提高浏览器响应速度和流畅度。本发明解决了传统智能终端App客户端需要频繁更新的缺点,具备B/S架构Web应用系统的兼容性和跨平台通用性,又解决了传统浏览器Web应用浏览器端功能以及实时交互上的不足,增强用户体验且能弥补智能终端计算能力不足的缺陷。



1. 一种改善Web浏览器应用效率的分布式协同方法,其特征在于,该方法包括:

通过一个运行于智能终端上的在智能终端浏览器后方的本地Web服务端模块与一个或多个远端Web服务模块,共同实现分布式协同计算的Web服务,并通过浏览器直接与本地Web服务端模块通信实现浏览器响应速度的提高;把本地Web服务端模块作为一种通用的中间件,使智能终端能以HTTP协议、XML文档为基础的自定义应用层通信方式与远端Web服务模块共同实现前述分布式协同计算;智能终端浏览器不直接与包括云端服务器在内的远端Web服务模块进行数据交换,而是与运行于后台的本地Web服务端模块进行数据交换,由本地Web服务端模块与各种远端Web服务模块进行数据交换;浏览器向该本地Web服务端模块发送请求,该本地Web服务端模块解析请求,并给浏览器所需的文件;浏览器通过AJAX及其他动态网页技术快速刷新部分页面,该本地Web服务端模块则通过HTTP协议、XML文档为基础的通信方式与包括云端服务器在内的远端Web服务模块进行数据交换,将用户请求转化为分布式计算任务的信息发给远端Web服务模块,获取布局文件并不断刷新本地布局文件,及时获取布局变化情况;

所述Web服务端模块运行于用户的智能终端后台,经由浏览器与用户交互,这个过程是遵循Http协议,通过网页文件进行的,本地Web服务端模块会成为协同计算的一个节点,与远端Web服务端模块进行数据交换,这个过程则是在遵循Http协议的基础上,通过特定的协议进行,该协议有XML文件的格式,远端Web服务模块之间的数据交换也遵循相同的协议;传输协议构建于标准的HTTP协议之上,直接获取文件的请求为GET,用于浏览器访问本地服务获取文件;本地Web服务端模块与远端Web服务模块交换数据以及远端Web服务模块与远端Web服务模块之间交换数据采用POST请求通过发送XML文件实现,内部标签包括但不限于ROLE、DIRECT、NAME、HREF、MODIFYTIME,EXTRAINFORMATION。

2. 根据权利要求1所述的一种改善Web浏览器应用效率的分布式协同方法,其特征在于,所述把本地Web服务端模块作为一种通用的中间件具体是:

在智能终端后台运行一个小型的本地Web服务端模块作为并将其作为通用的中间件;该本地Web服务端模块不包含用户界面,不涉及对底层硬件的控制;该本地Web服务端模块既能作为客户端向包括云端服务器在内的远端Web服务模块发送请求,也能作为本地Web服务端模块接收来自智能终端浏览器的请求;浏览器无法完成的计算与功能通过请求发送给该本地Web服务端模块,以充分利用智能终端硬件进行计算或与远端Web服务模块通信,实现网络上各节点的分布式计算;该本地Web服务端模块与本地浏览器之间基于HTTP协议交换数据,能返还给浏览器呈现用户界面所需的文件。

3. 根据权利要求1所述的一种改善Web浏览器应用效率的分布式协同方法,其特征在于,所述分布式协同计算中:本地浏览器不能与远端Web服务模块进行直接通讯,只能通过访问本地Web服务端模块来获取文件;各个角色之间都以HTTP协议进行通讯,本地浏览器与本地Web服务端模块之间是HTML通信,而本地Web服务端模块与远端Web服务模块之间以及远端Web服务模块与远端Web服务模块之间是以基于HTTP协议的自定义的统一的XML格式的协议进行通讯。

4. 根据权利要求1所述的一种改善Web浏览器应用效率的分布式协同方法,其特征在于本地Web服务端模块与远端Web服务模块进行数据交换的XML格式的协议包括:

格式标签包含以下信息:角色信息、发起请求的代号、目标指向地址、拓展信息、XML文

件发送时的时间戳,若有需要,通讯时能在拓展信息标签下增加所需标签,否则扩展信息标签内容为空。

一种改善Web浏览器应用效率的分布式协同方法

技术领域

[0001] 本发明涉及网络数据传输技术,特别涉及改善Web浏览器应用效率的分布式协同方法。

背景技术

[0002] 目前国内外在智能终端和远端Web服务程序的分布式协同计算方面,主要采用的方式有C/S架构(即App方式)、B/S架构(即浏览器方式)、SOA架构(即Web Service方式)。不过都存在某些不足。

[0003] (1)C/S架构(即App方式)的优点是交互能力优秀,缺点是对于用户来说需要频繁更新程序版本,非常麻烦,对于开发者来说,又需要适配不同硬件不同系统版本的智能终端,导致效率低下。

[0004] (2)B/S架构(即浏览器方式)的优点是更新在远端Web服务程序端完成,用户不需要下载App客户端,但此方法的交互性较差,响应有延迟,功能上不够强大,缺少主控,容易造成信息丢失等错误。

[0005] (3)SOA架构(即Web Service方式)太庞大臃肿,不适合直接部署到常见的智能终端尤其是移动智能终端上,移动终端硬件性能有限,Web Service容易运行缓慢,同样导致浏览器响应迟滞。

[0006] (4)直接通过浏览器进行许多操作时会带来控制权分散的问题,需要各个Web Server互相配合,没有主控点(浏览器不是主控)。这种设计,实现上较困难及容易出错。比如在一些交易场景下,可能会有故障导致交易中断导致金钱丢失的情况。

[0007] (5)虽然当今移动设备的硬件发展迅猛,软件优化加强,已经诞生了大量移动智能终端的实用软件,但是用户对硬件能力的需求是没有限制的,移动平台的硬件必须考虑体积、耗电、散热等等问题,无法无限地提高硬件性能,尤其由于原子直径的限制,现在的CPU性能已经快发展到一个瓶颈,更需要通过分布式协同计算的方式来提高处理能力。

发明内容

[0008] 本发明的目的在于克服现有技术存在的上述不足,提供了改善Web浏览器应用效率的分布式协同方法,具体技术方案如下。

[0009] 一种改善Web浏览器应用效率的分布式协同方法,该方法包括:

[0010] 通过一个运行于智能终端上的在智能终端浏览器后方的本地Web服务端模块与一个或多个远端Web服务模块,共同实现分布式协同计算的Web服务,并通过浏览器直接与本地Web服务端模块通信实现浏览器响应速度的提高;把本地Web服务作为一种通用的中间件,使智能终端能以HTTP协议、XML文档为基础的自定义应用层通信方式与远端Web服务模块共同实现前述分布式协同计算;智能终端浏览器不直接与包括云端服务器在内的远端Web服务模块进行数据交换,而是与运行于本地后台的Web服务端模块进行数据交换,由本地Web服务端模块与各种远端Web服务模块进行数据交换;浏览器向该本地Web服务端模块

发送请求,该本地Web服务端模块解析请求,并给浏览器所需的文件;浏览器通过AJAX及其他动态网页技术快速刷新部分页面,该本地Web服务端模块则通过HTTP协议、XML文档为基础的通信方式与包括云端服务器在内的远端Web服务模块进行数据交换,将用户请求转化为分布式计算任务的信息发给远端Web服务模块,获取布局文件并不断刷新本地布局文件,及时获取布局变化情况。

[0011] 进一步地,所述把本地Web服务作为一种通用的中间件具体是:

[0012] 在智能终端后台运行一个小型的本地Web服务端模块作为并将其作为通用的中间件;该本地Web服务端模块不包含用户界面,不涉及对底层硬件的控制;该本地Web服务端模块既能作为客户端向包括云端服务器在内的远端Web服务模块发送请求,也能作为本地Web服务端模块接收来自智能终端浏览器的请求;浏览器无法完成的计算与功能通过请求发送给该本地Web服务端模块,以充分利用智能终端硬件进行计算或与远端Web服务模块通信,实现网络上各节点的分布式计算;该本地Web服务端模块与本地浏览器之间基于HTTP协议交换数据,能返还给浏览器呈现用户界面所需的文件;该Web服务端模块与远端Web服务模块之间基于HTTP协议以自定义的XML格式的协议进行数据传输;另外,远端Web服务模块之间也是基于同样的协议进行数据传输。

[0013] 进一步地,所述分布式协同计算中:本地浏览器不能与远端Web服务模块进行直接通讯,只能通过访问本地Web服务端模块来获取文件;各个角色之间都以HTTP协议进行通讯,本地浏览器与本地Web服务端模块之间是HTML通信,而本地Web服务端模块与远端Web服务模块之间以及远端Web服务模块与远端Web服务模块之间是以基于HTTP协议的自定义的统一的XML格式的协议进行通讯。

[0014] 进一步地,本地Web服务端模块与远端Web服务模块进行数据交换的XML格式的协议包括:

[0015] 格式标签包含以下信息:角色信息、发起请求的代号、目标指向地址、拓展信息、XML文件发送时的时间戳,若有需要,通讯时能在拓展信息标签下增加所需标签,否则扩展信息标签内容为空。

[0016] 与现有技术相比,本发明具有如下优点和技术效果:

[0017] 本发明通过一个运行于智能终端上的,运行于智能终端浏览器后方的本地Web服务端模块与远端Web服务模块共同实现分布式的Web服务,能实现浏览器响应速度的提高并解决兼容性问题。相较于浏览器这种瘦客户端,该本地Web服务端模块能够根据实际需要进行更多的计算,并与远端Web服务模块共同实现分布式协同计算,更充分利用智能终端的性能并提升用户体验。该本地Web服务端模块依赖浏览器呈现用户界面但不依赖于特定的浏览器。该本地Web服务端模块能同时作为客户端和服务端存在,作为客户端能向远端Web服务模块发起请求,请求分布式运行应用以及获取相应的界面布局文件。该本地Web服务端模块能响应本地浏览器的请求,浏览器能通过向该本地Web服务端模块发送请求获取HTML文件以及其他与界面呈现相关的文件,如有需要,可通过AJAX等动态网页技术自动刷新部分页面。该本地Web服务端模块能作为客户端连接远端Web服务模块,在后台更新界面布局文件,由于传输的数据只有界面布局文件并且可以在后台传输,不需要调用浏览器,浏览器只用于呈现界面,故响应速度大幅提高,而且即便由于网络拥塞等问题造成了该本地Web服务端模块偶尔无法获取页面布局文件,也不会出现页面无法显示的情况,因为能及时返还

给浏览器已储存于本地的HTML文件;该软件接收XML文件后解析出界面布局文件并能自动调用浏览器显示文件。

[0018] 每个远端Web服务模块地位平等,能够作为客户端发起请求,也能作为服务器接收请求。各个Web服务端模块之间以XML文件,通过统一的格式进行通信。

[0019] 用户通过在浏览器上进行操作,将请求发给该本地Web服务端模块,该本地Web服务端模块则解析请求然后生成相应的符合格式的XML格式的请求,然后向远端Web服务模块发起请求,该本地Web服务端模块接收远端Web服务模块返还的界面布局文件后能自动调用浏览器显示文件。

[0020] 通过增加运行于后台的本地Web服务端模块,实现本地与远端Web服务模块的分布式协同计算,本地不需要像C/S架构一样频繁更新应用。通过浏览器来显示用户界面也解决了兼容性的问题。由于能通过不断获取本地页面文件来刷新部分页面,并且智能终端能及时从远端Web服务模块获取页面布局的更新情况,且不需要传输界面呈现所需的大部分图片,故能在不增加网络流量耗费的情况下显著提高响应速度。相对于B/S架构能显著提高交互能力。且该本地Web服务端模块可以较为方便地扩展功能,如对密码进行加密发送,断点上传下载等。这样就可以在不针对特定浏览器开发插件的情况下拓展浏览器的功能。另外,本发明部分逻辑类似SOA架构,但比SOA架构更加轻量化,只保留了需要的部分,使之能够部署到一般的智能终端包括移动智能终端上。

附图说明

[0021] 图1为本发明在交换数据的示意图;

[0022] 图2为本发明在智能终端运行的流程示意图。

[0023] 图3为本发明接收本地浏览器请求后的运行流程示意图。

具体实施方式

[0024] 为解决传统智能终端App客户端需要频繁更新的缺点,兼顾B/S架构Web应用系统的兼容性和跨平台通用性,解决Web应用浏览器端功能以及实时交互上的不足,增强用户体验且弥补智能终端计算能力不足的缺陷,以下提供一种改善Web浏览器应用效率的分布式协同方法,通过一个运行于智能终端本地的Web服务端模块来实现强于一般浏览器端的功能。

[0025] 如图1所示,该Web服务端模块运行于用户的智能终端后台,经由浏览器与用户交互,这个过程是遵循Http协议,通过网页文件进行的,本地Web服务端模块会成为协同计算的一个节点,与远端Web服务端模块进行数据交换,这个过程则是在遵循Http协议的基础上,通过特定的协议进行,该协议有XML文件的格式,远端Web服务模块之间的数据交换也遵循相同的协议。

[0026] 针对目前云端应用开发者需参照传输规范来进行数据链路层的开发,本发明的传输协议构建于标准的HTTP协议之上,直接获取文件的请求为GET,用于浏览器访问本地服务获取文件。

[0027] 该本地Web服务端模块与远端Web服务模块交换数据以及远端Web服务模块与远端Web服务模块之间交换数据采用POST请求通过发送XML文件实现,

[0028] 内部标签包括但不限于ROLE(内容是网络角色信息)、DIRECT(内容是目标动作)、NAME(内容是名称)、HREF(内容是网络路径)、MODIFYTIME(内容是请求文件的最后修改时间)、EXTRAINFO(内容是附加信息)。

[0029] 本发明公开的本地Web服务端模块运行的总体过程如图2所示,

[0030] 步骤201:本地Web服务端模块等待本地浏览器的请求;

[0031] 步骤202:本地Web服务端模块接收本地浏览器的请求,第一个请求一般为GET请求,浏览器通过该请求查看所能获取的请求的列表;

[0032] 步骤203:本地Web服务端模块返回列表网页文件给浏览器;

[0033] 步骤204:Web服务端模块接收并解析浏览器发来的请求,根据用户需要,开始向用户提供具体服务;

[0034] 步骤205:等待用户终止正在提供的服务;

[0035] 步骤206:只要用户没有终止服务,则不断生成符合要求的XML格式的请求;

[0036] 步骤207:通过用Http POST请求包装并向远端Web服务模块发送请求;

[0037] 步骤208:获取界面布局文件并更新存放在本地的界面布局文件,步骤207、208如果失败,则本地布局文件维持原样,等待下一次更新,浏览器获取到的依然是旧的文件而不会出现无法获取文件的情况;

[0038] 如果用户停止获取该服务,则转入步骤209。

[0039] 步骤209:按照规定格式生成终止服务应用的请求;

[0040] 步骤210:通过用Http POST请求包装并向远端Web服务模块发送请求。

[0041] 该本地Web服务端模块通过POST请求发送携带请求信息的XML文件,远端Web服务模块处理请求,返还相应的界面布局文件,该本地Web服务端模块可以接收该文件并调用浏览器显示,因为本发明中的Web服务模块并非现成的Web服务程序,而是基于底层的Socket编程,故有主动发起请求的能力。

[0042] 图3表述了本发明公开的Web服务端模块收到本地浏览器请求后的处理方式:

[0043] 步骤301:获取浏览器请求的具体内容;

[0044] 步骤302:判断浏览器需求,本地Web服务端模块与远程服务模块约定网页文件的标准写法,提交GET请求或通过POST请求发送获取文件的请求为获取文件的方法,若浏览器的请求为获取文件则前往步骤303,若为与远端Web服务模块的交互请求(包括获取服务,终止服务,提交数据等)则会跳往步骤304;

[0045] 步骤303:判断文件是否存在,并按照图示逻辑返回对应文件;

[0046] 步骤304:解析浏览器发送过来的请求并按照规定格式包装成XML格式请求;

[0047] 步骤305:返回给浏览器一个临时的loading页面,该页面会自动刷新,当本地Web服务端模块从远端Web服务模块获取到界面布局文件后会更新该文件;

[0048] 步骤306:将步骤304的请求发给远端Web服务模块。

[0049] 本地Web服务端模块存在于后台启动一个只面向本地浏览器的服务,本地浏览器通过访问该本地Web服务端模块获取文件。浏览器第一次发送获取应用的请求后该本地Web服务端模块会返送loading页面,然后会在后台更新文件,浏览器通过AJAX等动态网页技术快速刷新页面,以在第一时间更新页面。

[0050] 例如,在网络传输过程中,可能会经过5个以上的路由器,延时可能达到100ms,中

间可能出现100K的带宽瓶颈,这样直接使用浏览器来访问远端Web服务模块,显示就会容易出现迟钝,影响用户体验,而使用本地Web服务端模块与远端Web服务模块进行通讯并获取文件后,浏览器即可与该本地Web服务端模块通讯来获取相应文件,本地虚拟带宽在1G以上,传输延时低于1ms,这样用浏览器来显示页面就不会出现响应迟钝的情况。另外,该本地Web服务模块可以对数据进行技术处理,类似于流媒体播放器的缓存,使得浏览器的显示更加流畅。

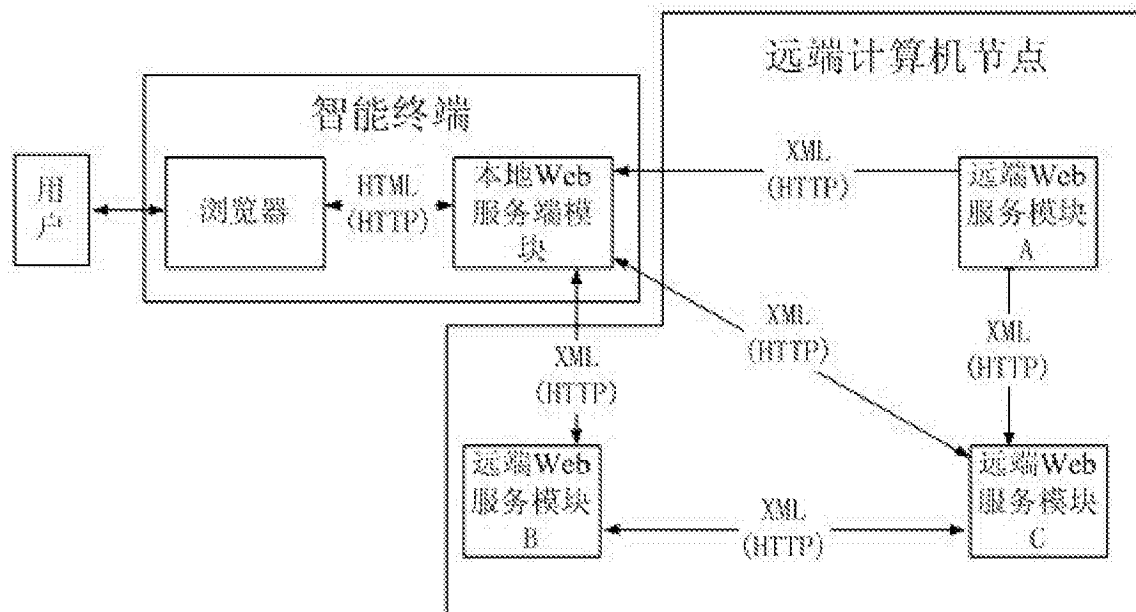


图1

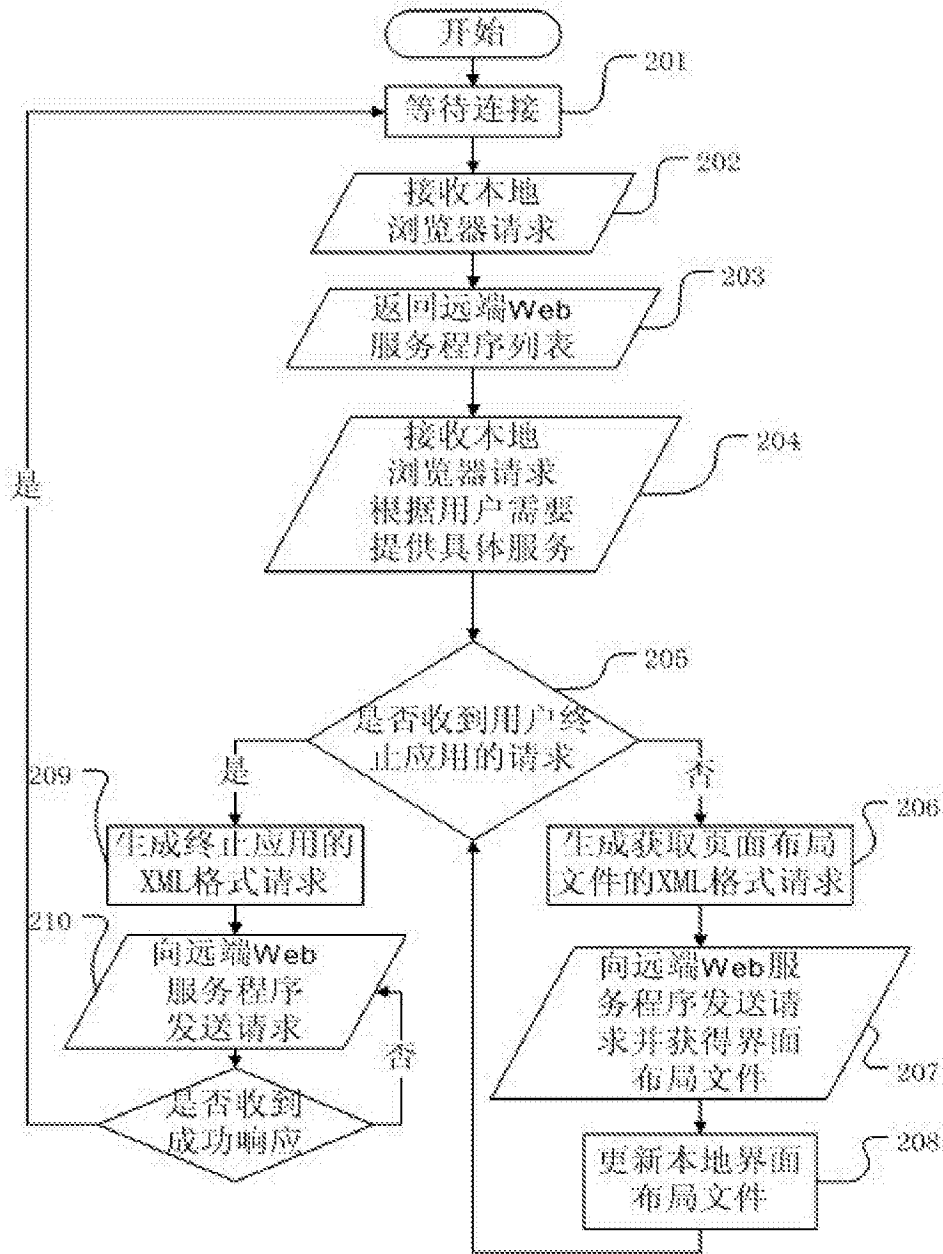


图2

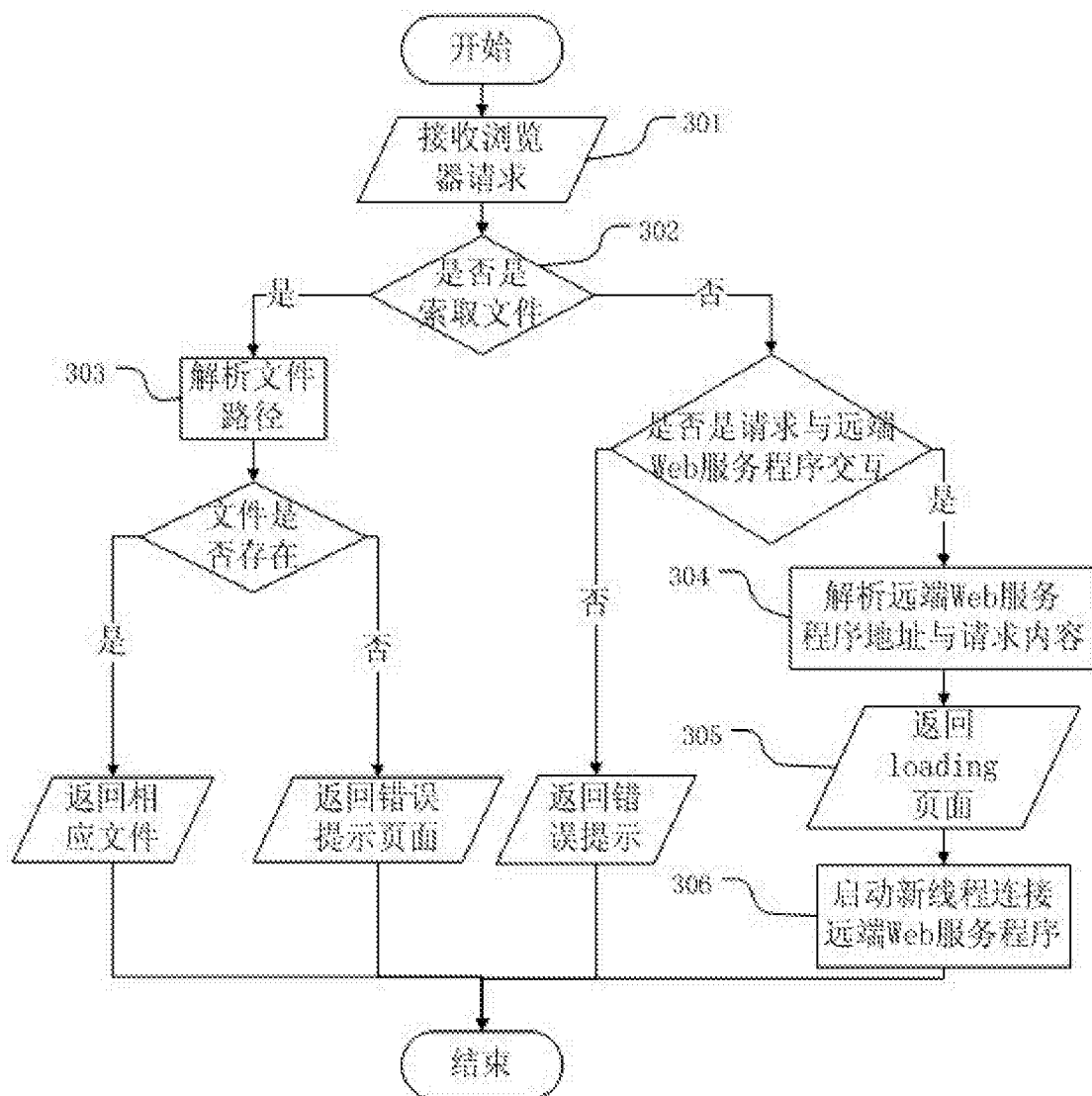


图3