



## (12)发明专利

(10)授权公告号 CN 104298511 B

(45)授权公告日 2018.08.03

(21)申请号 201410529630.8

(22)申请日 2014.10.10

(65)同一申请的已公布的文献号

申请公布号 CN 104298511 A

(43)申请公布日 2015.01.21

(73)专利权人 王钟

地址 266045 山东省青岛市四方区商邱路  
10号2单元504

(72)发明人 王钟

(74)专利代理机构 北京德琦知识产权代理有限公司

公司 11018

代理人 王民盛 王丽琴

(51)Int.Cl.

G06F 8/65(2018.01)

G06F 9/445(2018.01)

(56)对比文件

US 2008040426 A1,2008.02.14,

CN 102037710 A,2011.04.27,

US 2008040426 A1,2008.02.14,

CN 102037710 A,2011.04.27,

CN 101895531 A,2010.11.24,

US 2011138354 A1,2011.06.09,

CN 103927163 A,2014.07.16,

审查员 吴阳

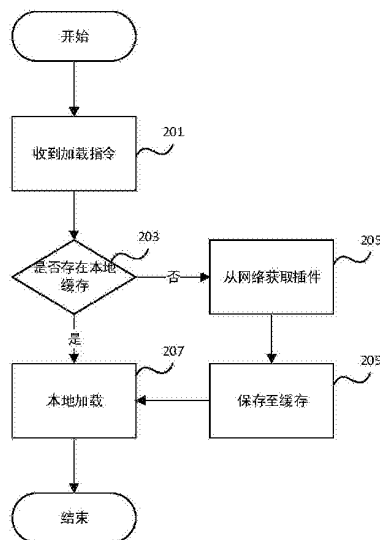
权利要求书1页 说明书3页 附图2页

(54)发明名称

实现网络化远程插件的方法及系统

(57)摘要

本申请公开了一种网络化远程插件系统,包括:本地主机、网络和远程主机;所述本地主机上安装有一个软件系统,该软件系统包含:插件本地代理主办模块121和插件本地代理109;所述插件本地代理主办模块121提供加载和运行时环境和管理插件本地代理109的功能;所述网络用于本地主机上的插件本地代理主办模块121从远程主机下载插件程序,以及用于插件本地代理109向远程主机发送消息处理请求和获得处理结果回应。本申请还公开了一种通过网络加载插件的实现方法和一种通过网络远程运行插件的实现方法。本发明把插件技术与网络化结合,改造插件加载和运行的过程和方式,使插件具有:程序结构的相对动态可变性、功能相对多样性以及更新容易等特点。



1. 一种网络化远程插件系统,其特征在于,包括:本地主机、网络和远程主机;

所述本地主机上安装有一个软件系统,该软件系统包含:插件本地代理主办模块121和插件本地代理109;

所述插件本地代理主办模块121提供加载和运行时环境和管理插件本地代理109的功能;

所述插件本地代理109是由本地代理主办模块121加载的插件程序;

所述网络用于本地主机上的插件本地代理主办模块121从远程主机下载插件程序,以及用于插件本地代理109向远程主机发送消息处理请求和获得处理结果回应;

其中,所述插件代指一种为任意软件系统提供该系统功能扩展、访问或修改系统的程序,只为系统提供功能而没有访问或修改系统功能则不属于所述插件。

2. 一种通过网络远程运行插件的实现方法,其特征在于,包括:

插件本地代理109收到消息、事件或功能调用;

插件本地代理109将该消息、事件或功能调用的细节或参数转发到网络,网络上的远程主机在进行处理响应后发送回应的处理结果数据;

插件本地代理109接收到网络回应的处理结果数据,如果回应数据是程序或命令类型则运行该程序或执行该命令,如果是纯数据类型则交由在本地消息处理机制继续处理完成;

其中,所述插件代指一种为任意软件系统提供该系统功能扩展、访问或修改系统的程序,只为系统提供功能而没有访问或修改系统功能则不属于所述插件。

## 实现网络化远程插件的方法及系统

### 技术领域

[0001] 本申请涉及计算机软件领域的插件技术,特别涉及一种实现网络化远程插件的方法及系统。

### 背景技术

[0002] 插件(又译外挂,英文为Plug-in、addin、add-in、addon或add-on或app,)是一种电脑程序,通过和应用程序(例如网页浏览器,电子邮件服务器)的互动,用来替应用程序增加一些所需要的特定的功能。最常见的有游戏、网页浏览器的插件和媒体播放器的插件。(引自维基百科“插件”词条)

[0003] 现有技术中,插件(plugin)或扩展(extension)在本质上是本地程序,没有类似网页的动态性,其程序结构相对固定,功能相对单一。

[0004] iSCSI虚拟磁盘接口,就是在本地电脑里面虚拟出一个远程电脑里面的磁盘。现有技术中,iSCSI虚拟磁盘通过网络收发硬盘接口指令,通过软件在本地虚拟了硬件插件,实现了硬件插件的网络化。但其全部技术局限于模拟硬件。

### 发明内容

[0005] 在本申请中,名词“插件”含义指不仅限于指应用程序的插件,而是代指一种为任意软件系统提供该系统功能扩展、访问或修改系统的程序。只为系统提供功能而没有访问或修改系统则不能算为插件,比如:浏览器内运行显示的网页不能算浏览器的插件。

[0006] 本申请提供了一种网络化远程插件的系统,包括:本地主机、网络和远程主机;

[0007] 所述本地主机上安装有一个软件系统,该软件系统包含:插件本地代理主办模块121和插件本地代理109;

[0008] 所述插件本地代理主办模块121提供加载和运行时环境和管理插件本地代理109的功能;

[0009] 所述网络用于本地主机上的插件本地代理主办模块121从远程主机下载插件程序,以及用于插件本地代理109向远程主机发送消息处理请求和获得处理结果回应。

[0010] 本申请还提供了一种通过网络加载插件的方法,包括:

[0011] 插件本地代理主办模块121收到加载插件的消息;

[0012] 插件本地代理主办模块121从网络获取下载插件;

[0013] 插件本地代理主办模块121加载插件。

[0014] 本申请还提供了一种通过网络远程运行插件的实现方法,包括:

[0015] 插件本地代理109收到消息、事件或功能调用;

[0016] 插件本地代理109将该消息、事件或功能调用的细节或参数转发到网络,网络上的远程主机在进行处理响应后发送回应;

[0017] 插件本地代理109接收到网络回应的处理结果数据,如果回应数据是程序或命令类型则运行该程序或执行该命令,如果是纯数据类型则交由在本地消息处理机制继续处理

完成。

[0018] 网络化远程插件技术方案相比于现有技术差别在于：加载网络化，计算执行远程化。

[0019] 加载网络化，即加载过程借助本地软件系统宿主从网络或本地缓存中加载。

[0020] 计算执行远程化，即计算处理的过程转移至网络完成，执行的操作是通过把数据解析为命令，在本地执行完成。这里网络可以是在网络上的另外一台或几台主机，或者也可以通过网络分布式计算。

[0021] 本发明把插件技术与网络化结合，改造插件加载和运行的过程和方式，使插件具有：程序结构的相对动态可变性、功能相对多样性以及更新容易等特点。但是同时，由于依赖网络相对丧失部分独立性。

### 附图说明

[0022] 图1为本申请实施例提供的网络化远程插件系统示意图；

[0023] 图2为本申请实施例提供的网络化远程插件加载流程图；

[0024] 图3为本申请实施例提供的网络化远程插件处理执行过程流程图。

### 具体实施方式

[0025] 以下结合图示进一步说明技术原理。

[0026] 以下所述内容都是对本发明的举例说明。为了方便说明、帮助理解，突出了特征。对现有技术和非特征部分，在不影响理解的地方，进行省略简化。切不可把本发明的实施例当作本发明的约束限制来理解。

[0027] 图1为本申请实施例提供的网络化远程插件系统示意图。其中包含：本地主机101、本地软件系统105、网络111、插件远程服务器103、插件本地代理主办模块121、插件本地代理109和缓存115。

[0028] 本地主机101可以是任何一种计算机，比如像：超级计算机、微型PC、膝上电脑，智能手机等。

[0029] 本地软件系统105可以是任何一种软件系统，比如：操作系统、应用程序等。

[0030] 网络111可以是任何一种计算机网络。

[0031] 插件远程服务器103是在网络远程的一台服务器，提供插件程序下载和针对网络化远程插件的计算等服务。

[0032] 插件本地代理主办模块121提供是加载和运行时环境和管理插件本地代理109的功能和职责。最简单的运行时环境可以是解析单个单词或单个字母为命令并执行比如：DOS命令行；一个复杂的运行时环境可以提供完善运行时库和操作系统的编程接口的高级编程语言虚拟机或脚本宿主，比如：java虚拟机环境，javascript引擎，vbscript脚本宿主等。对插件本地代理109的管理可以包括启动加载、暂停使用、恢复使用、关闭终止等。

[0033] 插件本地代理109是由本地代理主办模块121加载的插件程序。

[0034] 缓存115，既对插件本地代理109程序的永久化或运行时临时性本地缓存。

[0035] 图2为本申请实施例提供的网络化远程插件加载流程图。包括：

[0036] 步骤201：插件本地代理主办模块121收到加载某款插件的消息。

[0037] 步骤203:插件本地代理主办模块121检查该款插件是否存在于本地缓存115中,若是,执行步骤207,否则执行步骤205。

[0038] 步骤205:插件本地代理主办模块121从网络获取下载该插件。

[0039] 步骤207:插件本地代理主办模块121从本地缓存115中提取并加载该款插件。

[0040] 在本步骤中,加载的意思是最终载入一个可以正常运行、与系统兼容的插件程序。根据不同的实施例,加载过程可能需要包含额外其它环节。比如:如果插件代码是源代码形式的则需要先编译等。

[0041] 步骤209:下载下来的插件被保存至本地缓存。

[0042] 图3为本申请实施例提供的网络化远程插件处理执行过程流程图。

[0043] 步骤301:插件本地代理109收到一个消息、事件或功能调用

[0044] 步骤303:插件本地代理109判断该事件、消息或调用的功能是否需要通过网络完成。若是,执行步骤305,否则执行步骤307。

[0045] 步骤305:插件本地代理109将该消息、事件或功能调用的细节或参数转发到网络。在本实施例中插件远程服务器103将接收、处理并做出回应。

[0046] 步骤307:插件本地代理109使用本地的消息处理机制处理该消息。如果该消息经过网络处理含有网络处理结果,插件本地代理109将检查处理结果。如果处理结果是数据类型(非程序)则利用此结果在本地进一步处理。如果处理结果是程序类型,插件本地代理109会执行该程序。

[0047] 在本步骤中,根据不同的实施例,执行过程可能需要包含额外其它环节。如果程序类型的处理结果是源代码或脚本的则需要先编译或解析。根据不同的实施例,执行中的程序可能向网络再次发出新的消息请求处理,即产生递归,多线程等情况。

[0048] 步骤309:插件本地代理109等待并收到来自网络回应的处理结果。

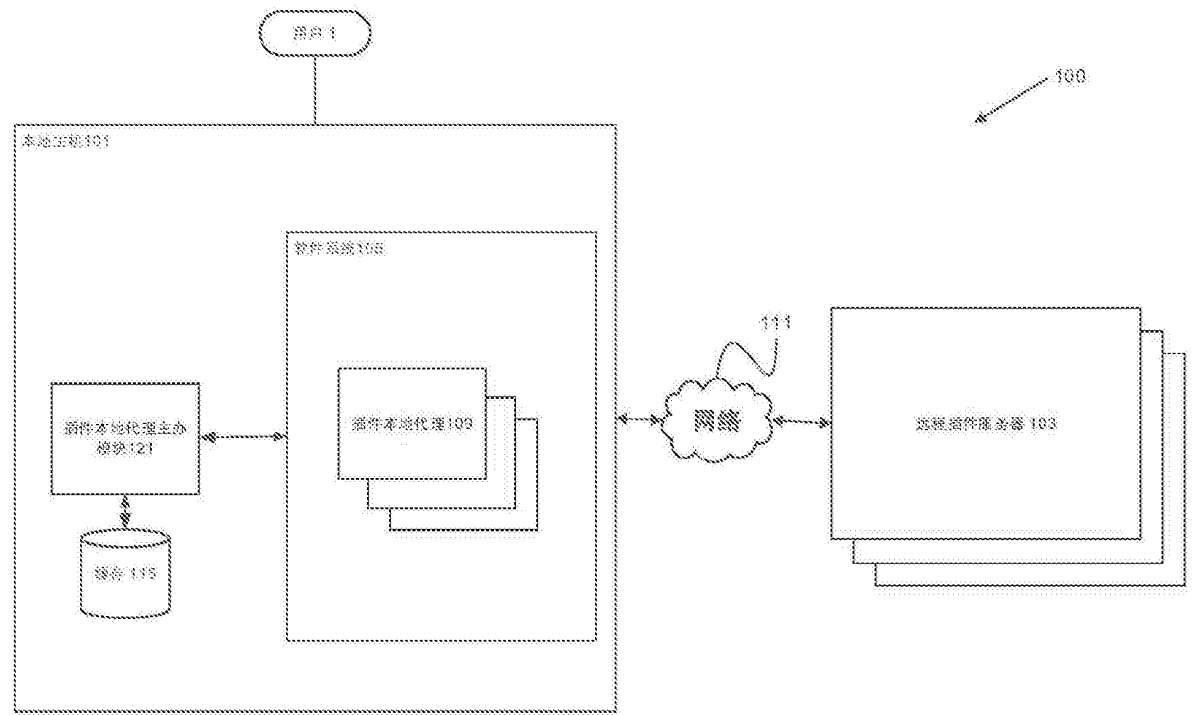


图1

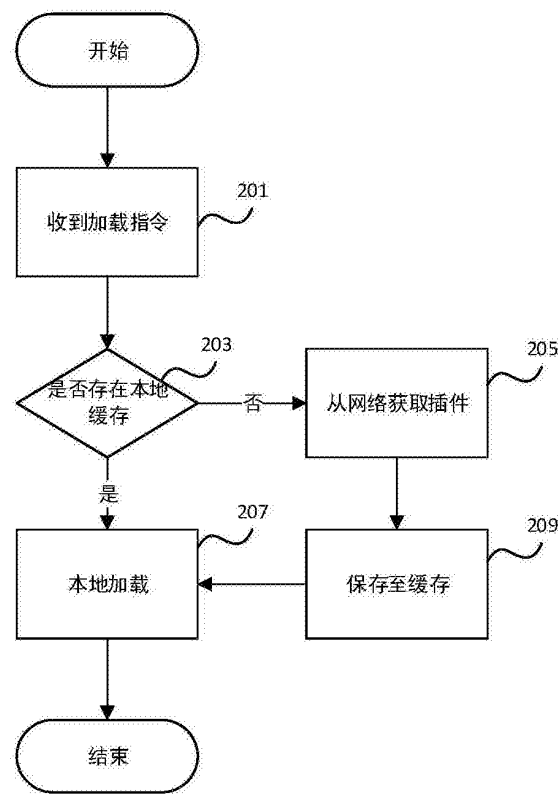


图2

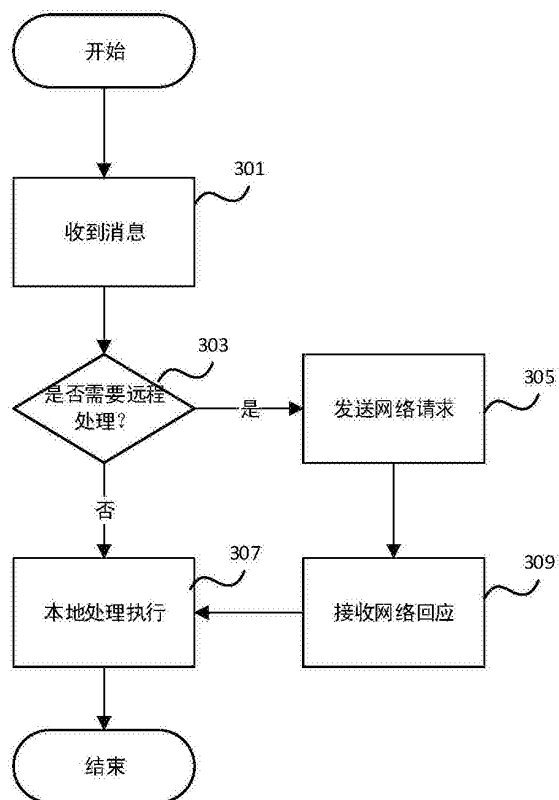


图3