



(12)发明专利

(10)授权公告号 CN 103634138 B

(45)授权公告日 2016.12.28

(21)申请号 201210308284.1

H04L 29/08(2006.01)

(22)申请日 2012.08.27

(56)对比文件

(65)同一申请的已公布的文献号

CN 102427466 A, 2012.04.25,

申请公布号 CN 103634138 A

CN 102427466 A, 2012.04.25,

(43)申请公布日 2014.03.12

CN 101201617 A, 2008.06.18,

(73)专利权人 阿里巴巴集团控股有限公司

CN 101154106 A, 2008.04.02,

地址 英属开曼群岛大开曼资本大厦一座四
层847号邮箱

EP 1746764 A2, 2007.01.24,

(72)发明人 黄峤睿

CN 101571876 A, 2009.11.04,

(74)专利代理机构 北京国昊天诚知识产权代理
有限公司 11315

审查员 王灿

代理人 许志勇

(51)Int.Cl.

H04L 12/24(2006.01)

权利要求书2页 说明书6页 附图3页

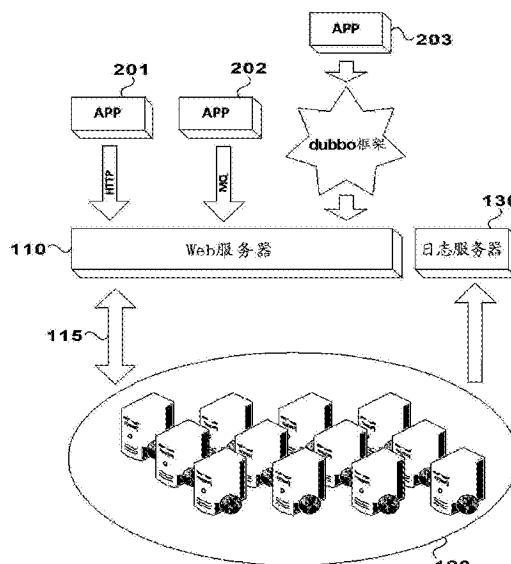
H04L 29/06(2006.01)

(54)发明名称

分布式调度的远程管理与运维方法及其系
统

(57)摘要

本发明公开了一种分布式调度的远程管理及运维方法,包括部署web服务器,且当web服务器接收来自客户端的任务描述时,建立该web服务器与受控服务器之间的传输安全协议;该受控服务器接收该web服务器所转发的任务描述,并根据该任务描述执行相对应的脚本。本发明借由web服务器的中控操作,实现了兼具胖client与瘦client优势的分布式C/S系统架构,且web服务器提供了友好的编程接口,方便上层应用的开发与功能扩展。



1. 一种分布式调度的远程管理及运维方法,其特征在于,包括下列步骤:

a) 部署web服务器;

b) 当该web服务器接收来自客户端的任务描述时,建立该web服务器与受控服务器之间的传输安全协议;以及

c) 该受控服务器接收该web服务器所转发的该任务描述;受控服务器自web服务器上下载代理与任务编号;代理根据该任务编号到web服务器上查找与该任务编号相对应的详细信息,包括前置任务、脚本名称及脚本参数其中之一;代理下载并执行相对应的脚本。

2. 根据权利要求1所述的方法,其特征在于,还包括:步骤d)响应于该脚本的执行,将所执行的脚本的详细信息作为日志并输出至日志服务器进行归类与储存。

3. 根据权利要求1所述的方法,其特征在于,步骤b)所建立的传输安全协议是建立在应用层与传输层基础上的SSH传输协议。

4. 根据权利要求1所述的方法,其特征在于,客户端的任务描述是以应用的形式、通过特定网络传输协议发送给web服务器。

5. 根据权利要求4所述的方法,其特征在于,该特定网络传输协议是超文本传输协议,且该应用的编程内容不包含与建立该web服务器与该受控服务器之间的传输安全协议有关的编程。

6. 根据权利要求1所述的方法,其特征在于,步骤c)还包括下列步骤:

c1) 通过脚本模板引擎将web服务器上的脚本模板渲染成用于执行的脚本,以供代理下截并执行该脚本。

7. 根据权利要求2所述的方法,其特征在于,步骤d)还包括下列步骤:

d1) 以正则表达式对所输出的日志进行比对,当所比对的输出满足特定格式时,即触发该输出所对应API的调用。

8. 一种分布式调度的远程管理及运维系统,其特征在于,包括:

web服务器,用于接收并转发来自客户端的任务描述;以及

受控服务器,用于接收转发自该web服务器的任务描述并根据该任务描述自该web服务器上下载代理与任务编号,且代理根据该任务编号到web服务器上查找与该任务编号相对应的详细信息,并下载、执行相对应的脚本,以及,响应于该脚本的执行,将其执行的脚本的详细信息作为日志输出。

9. 根据权利要求8所述的系统,其特征在于,所述系统还包括:

日志服务器,用于接收该受控服务器输出的日志,并进行归类与储存,

其中,该web服务器接收来自客户端的任务描述时,建立该web服务器与该受控服务器之间的传输安全协议。

10. 根据权利要求8所述的系统,其特征在于,该web服务器与该受控服务器之间的传输安全协议是建立在应用层与传输层基础上的SSH传输协议。

11. 根据权利要求10所述的系统,其特征在于,客户端的任务描述是以应用的形式、通过特定网络传输协议发送给web服务器。

12. 根据权利要求11所述的系统,其特征在于,该特定网络传输协议是超文本传输协议,且该应用的编程内容不包含与建立该web服务器与该受控服务器之间的传输安全协议有关的编程。

13.根据权利要求9所述的系统，其特征在于，该web服务器中还包含脚本模板引擎，用于将web服务器上的脚本模板渲染成用于执行的脚本，以供代理下载并执行该脚本。

分布式调度的远程管理与运维方法及其系统

技术领域

[0001] 本发明涉及服务器通讯领域,尤其涉及一种分布式调度的远程管理与运维方法及其系统。

背景技术

[0002] 客户端-服务器(C/S)架构是一种典型的网络架构,其中,服务器是一种高性能的计算机,能提供各种共享服务,例如文件、Web应用、数据库、文件邮件、打印及其它高性能的应用。然而当运维人员开始针对C/S架构进行管理或相关操作时,往往存在需要通过各种不同的运维工具直接登录并控制服务器的需求,造成服务器安全管理上的困难产生。再者,在C/S架构的日常运维的过程中,总有类似打包、批量部署应用程序、备份和还原数据库等需求,当服务器达较多的时候,就不得不借助一些工具补助。而这些工具都有的一个共同特点就是运行命令和脚本。

[0003] 一般来说,在C/S网络架构运维操作时的命令或脚本的具体运行方式可分为两大类,第一类称为瘦client(客户端)式,主要特色就是通过SSH(Secure Shell安全外壳)证书信任直接登录到服务器,以执行在服务器上所配置的各命令或脚本。另一类则是胖client式,主要特色是预先在各服务器上部署代理,由中控服务器发送命令给各服务器上的代理,并由代理执行各种操作,有的服务器是从中心脚本服务器下载脚本执行,有的服务器则是执行本机预部署的脚本,但无论如何都需要由胖client对各服务器完成代理的部署才能执行各项操作。

[0004] 此外,典型的C/S网络架构可适用各种不同操作系统或不同运维工具间的兼容操作。然而,当各种运维工具都有远程操作服务器的需求,但是同样的需求却要在不同系统或不同工具上不断地重复执行时,这可能会造成系统本身很大的资源浪费。另外,各类运维工具在运行脚本任务时,都需要长时间地与服务器开启一个SSH会话,并且需要单独打开一个线程保持状态,一旦线程崩溃时,前面的工作将会付诸东流。

[0005] 因此,面对前述的各项缺失,无论是瘦client还是胖client的部署方式,都需要在服务器上进行预配置或预部署,要么是全套功能的脚本集,要么是功能强大的代理,但这些都克服不了典型C/S架构的存在经典问题:1.很难实现快速部署安装和配置;以及2.客户端升级和维护的成本巨大。为了克服前述问题,本发明设计了一种分布式调度的远程管理与运维方法与系统。

发明内容

[0006] 本发明的主要目的在于提供一种分布式调度的远程管理及运维方法,以解决现有技术存在的不易实现远程控制程序的快速部署安装和配置以及客户端升级和维护的成本巨大等问题,其中本发明所提出的分布式调度的远程管理及运维方法,包括下列步骤:

[0007] 部署web服务器;

[0008] 当所述web服务器接收来自客户端的任务描述时,建立所述web服务器与受控服务

器之间的传输安全协议;以及

[0009] 所述受控服务器接收所述web服务器所转发的任务描述,并根据所述任务描述执行相对应的脚本。

[0010] 根据本发明的实施例,响应于所述脚本的执行,将所执行的脚本的详细信息作为日志并输出至日志服务器进行归类与储存。

[0011] 根据本发明的实施例,本发明所述的分布式调度的远程管理及运维方法中所述web服务器与所述受控服务器之间的传输安全协议所建立的传输安全协议是建立在应用层与传输层基础上的SSH传输协议。

[0012] 根据本发明的实施例,本发明所述的分布式调度的远程管理及运维方法中,更包含所述客户端的任务描述是以应用的形式、通过特定网络传输协议发送给web服务器。

[0013] 根据本发明的实施例,本发明所述的分布式调度的远程管理及运维方法中,所述特定网络传输协议是超文本传输协议,且所述应用的编程内容不包含与建立web服务器该受控服务器之间的传输安全协议有关的编程。

[0014] 根据本发明的实施例,本发明所述的分布式调度的远程管理及运维方法中,还包括:

[0015] 使所述受控服务器自web服务器上下载代理与任务编号;

[0016] 所述代理根据所述任务编号到web服务器上查找与任务编号相对应的详细信息,包括前置任务、脚本名称及脚本参数其中之一;以及

[0017] 代理下载并执行相对应的脚本。

[0018] 根据本发明的实施例,本发明所述的分布式调度的远程管理及运维方法中,包括通过脚本模板引擎将web服务器上的脚本模板渲染成用于执行的脚本,以供代理下载并执行所述脚本。

[0019] 根据本发明的实施例,本发明所述的分布式调度的远程管理及运维方法中,还包括通过正则表达式对所输出的日志进行比对,当所比对的输出满足特定格式时,即触发所对应API的调用。

[0020] 本发明还提供了一种分布式调度的远程管理及运维系统,包括:

[0021] web服务器,用于接收并转发来自该客户端的任务描述;以及

[0022] 受控服务器,用于接收转发自该web服务器的任务描述并根据该任务描述执行相对应的脚本,并且响应于该脚本的执行,将其执行的脚本的详细信息作为日志输出。

[0023] 根据本发明的实施例,本发明所述的分布式调度的远程管理及运维系统还包括日志服务器,用于接收该受控服务器输出的日志,并进行归类与储存,

[0024] 其中,该web服务器接收来自客户端的任务描述时,建立该web服务器与该受控服务器之间的传输安全协议。从而,可以以让该受控服务器开始执行该内务描述所对应的执行脚本。

[0025] 根据本发明的实施例,本发明所述的分布式调度的远程管理及运维系统中,所述web服务器与所述受控服务器之间的传输安全协议是建立在应用层与传输层基础上的SSH传输协议。

[0026] 根据本发明的实施例,本发明所述的分布式调度的远程管理及运维系统中,所述客户端的任务描述是以应用的形式、通过特定网络传输协议发送给web服务器。

[0027] 根据本发明的实施例，本发明所述的分布式调度的远程管理及运维系统中，所述特定网络传输协议是超文本传输协议，且所述应用的编程内容不包含与建立web服务器与该受控服务器之间的传输安全协议有关的编程。

[0028] 根据本发明的实施例，本发明所述的分布式调度的远程管理及运维系统中，所述受控服务器自所述web服务器上下载代理与任务编号，且代理根据所述任务编号到web服务器上查找与任务编号相对应的详细信息，并且下载、执行相对应的脚本。

[0029] 根据本发明的实施例，本发明所述的分布式调度的远程管理及运维系统中，所述web服务器中更包含脚本模板引擎，用于将web服务器上的脚本模板渲染成用于执行的脚本，以供代理下载并执行该脚本。

[0030] 与现有技术相比，根据本发明的技术方案，在每次脚本运行时才从web服务器下载最新版的代理运行，同时所有需要运行的脚本都是从web服务器下载，所以不存在预部署大量脚本的工作。另外，本发明对于受控服务器的要求仅是部署一个受认证的公钥，因此可以说是零成本的部署。在不对原有的环境进行任何改动的情况下，仅借由提供web服务器即实现了对整个环境的远程管理，既无需纠缠于SSH远程连接的安全性问题，同时也无须关心脚本的部署问题，提高了软件质量和交付速度，也有效做到了脚本的集中存放和统一版本管理。

附图说明

[0031] 此处所说明的附图用来提供对本发明的进一步理解，构成本申请的一部分，本发明的示意性实施例及其说明用于解释本发明，并不构成对本发明的不当限定。在附图中：

[0032] 图1是本发明的一种分布式调度的远程管理及运维系统实施例的整体架构图；

[0033] 图2是本发明的一种分布式调度的远程管理及运维系统实施例的系统架构与外部系统的相互关系示意图；以及

[0034] 图3是本发明的一种分布式调度的远程管理及运维方法实施例的流程示意图。

具体实施方式

[0035] 本发明的主要思想在于提供一种分布式调度的远程管理及运维方法及系统，以克服C/S网络架构中很难实现快速部署安装和配置，以及客户端升级和维护的成本巨大等传统缺失。为使本发明的目的、技术方案和优点更加清楚，以下结合附图及具体实施例，对本发明作进一步地详细说明。

[0036] 请参考图1，图1是本发明实施例的一种分布式调度的远程管理及运维系统100的整体架构图，主要包括由一台或数台web服务器111、112等所构成的web服务器群集110，以及至少一台具特定功能的受控服务器120。

[0037] 其中，例如web服务器111接收来自客户端的任务描述。根据本发明的实施例，客户端的任务描述是以应用的形式、通过特定网络传输协议115发送给web服务器，其中特定网络传输协议是超文本传输协议(http)，且该应用的编程内容不包含与建立该web服务器与该受控服务器之间的传输安全协议有关的编程。

[0038] 然后，所述受控服务器120接收来自所述web服务器111所发送的任务描述时，可自所述web服务器111下载对应的任务脚本，以执行相对应的脚本。更具体地说，受控服务器

120自web服务器111上下载代理与任务编号;然后代理根据该任务编号到web服务器111上查找与该任务编号相对应的详细信息,包括前置任务、脚本名称及脚本参数其中之一;以及通过脚本模板引擎将web服务器上的脚本模板渲染成用于执行的脚本,以供代理下载并执行该脚本。

[0039] 系统100还可以包括日志服务器130。响应于脚本的执行,所述受控服务器120可将执行的脚本的详细信息作为日志,输出到日志服务器130(如图1中右侧箭头所示),进行必要的归类及储存。例如,可以以正则表达式对所输出的日志进行比对,当所比对的输出满足特定格式时,即触发该输出所对应API的调用。

[0040] 本发明的主要的核心构思之一是在不对原有环境进行任何更动的情况下,借由部署一台或数台Web服务器111来实现对整个网络环境的远程管理。所述Web服务器111的具体功用在于替代传统的客户端来建立与特定功能的受控服务器120的联机,而避免客户端必须长时间直接登录所述受控服务器而造成服务器主机安全管理上的负担。前述具有特定功能的受控服务器120在本具体实施例中,本身可以是虚拟服务器或物理服务器等具备如文件应用、Web应用、数据库、文件邮件、打印等高性能应用的服务器。本实施例中,借由建立受控服务器120与web服务器群集110之间的传输安全协议115,web服务器群集110的一台或数台web服务器111可通过所建立的传输安全协议115对所述受控服务器120进行登录、访问、部署应用程序、备份和还原数据库等必要控制。在本具体实施例中,前述的传输安全协议可以是建立在应用层与传输层基础上的SSH传输协议。

[0041] 在实际中,由一台或数台web服务器111所构成的web服务器群集110可以视为这些受控服务器120的中控服务器。但考虑到一般web服务器未必具备集群功能,为了均衡服务器群集内的各web服务器111的负载,可借助如NGINX等类型的服务器来均衡各web服务器111的负载。

[0042] 请继续参考图2,图2是本发明的一种分布式调度的远程管理及运维系统与外部系统的相互关系示意图。下面将进一步借由图2的相互关系图说明本发明分布式调度的远程管理及运维系统各部件的详细运作。

[0043] 本发明的web服务器群集110在C/S网络架构中充当了所有受控服务器120与客户端交互的中间件,不仅有效地把控这些服务器120的安全,同时也提供了多种编程接口的功能。如图2所示,一般情况下,各客户端通常是利用如应用(APP)201、202、203的形式来对受控服务器120进行任务指派。尽管这些APP 201、202、203可能是来自不同作业环境的客户端,但不论是什么作业环境,只要这些客户端上可以执行基于web服务器110所开发的APP,并且通过各客户端自身所支持的网络传输协议115,例如Http、MQ或Dubbo等建立与web服务器110之间的网络连接,那么这些客户端的APP开发者便可以在不用关心受控服务器的连接与操作环境等细节上,专心的将精力投注在业务逻辑的开发上。因此,借由本发明的主要构思精神之一的web服务器110的部署,可以实现了APP和受控服务器之间的连接关系脱耦,所开发的APP不但不需要支持打开并保持一个长时间的SSH的会话连接,而只需由web服务器负责呼叫回传,可说是实现了高效的异步调用,同时也支持同步轮询等待等功能。

[0044] 另一方面,根据本发明所实现的分布式调度的远程管理及运维系统100,借由提供web服务器110的部署,能够提供各种不同的编程接口,这使得本发明web服务器110也能适用于动态脚本的开发。因为在一般情况下,在日常运维的过程中,通常是不止维护一套环

境,但其实针对不同环境所需脚本却仅仅是小部分存在差异。因此,本发明的另一个核心构想是将web服务器中所有脚本都作为“模板”对待,仅在下载时先经过模板引擎的渲染生成用于执行的脚本。

[0045] 由于动态脚本的出现,脚本的编写可以变得非常灵活,例如下面是一组可以将web服务器(例如,可以命名为salmon)的相关配置写入脚本中的一个脚本片断:

```
[0046] self.app_predicate="{{salmon_app_predicate}}"
[0047] self.log_server="{{salmon_client_syslog_address}}"
[0048] self.log_server_port="{{salmon_client_syslog_port}}"
[0049] self.task_server="{{salmon_task-server}}"
```

[0050] 如上所述,借由模板的设计,一般编程人员在开发如系统配置的相关程序时,只需要把脚本上的差异当成是具体参数的变换,就可以完成将特定服务器的配置带入到脚本中的目的,如此一来,编程人员只要先做好脚本模板的编辑工作后,后续的编程复杂度就便大幅降低,因此可以有效降低人力资源的浪费。

[0051] 本发明的分布式调度的远程管理及运维系统除了前述特点外,另一个可以达成的技术特点则是在受控服务器执行指派任务的相关脚本时,也可以一并实现对web服务器110上的特定应用程序接口(API)的调用。一般来说,在运维工具在受控服务器上执行相关任务时,通常是让受控服务器从胖client端下载代理或借由直接在受控服务器上预部署代理的方式,执行相关的脚本,已完成指派的任务。但是对于传统C/S网络架构来说,这些借由下载或预部署的代理并不具备调用运维工具API的能力,因此不可能在执行指派的任务命令时,又可以同时调动运维工具的特定API,例如打印功能API等。

[0052] 根据本发明的分布式调度的远程管理及运维系统,由于系统中的受控服务器会将每一个执行脚本的输出流以系统日志(syslog)的方式发送到日志服务器(如图2中右侧箭头所示),因此,只要可以针对这些脚本的输出流以高效的方式进行比对,就可以轻易借由这些脚本输出流来触发特定的API。在本发明的具体实施例中,借由正则表达式来比对所述脚本输出流的逐行输出,一旦比对到满足特定格式的输出即触发API的调用。如此一来,便可以轻易达成调用运维工具API的目的。

[0053] 请继续参阅图3,图3是本发明的一种分布式调度的远程管理及运维方法实施例的步骤执行示意图。根据本发明的另一核心构思之一,就是提供明的一种分布式调度的远程管理及运维方法,包括下列步骤:

[0054] 首先,在传统的C/S网络架构中部署web服务器,接收来自客户端的任务描述,如图3的CreateTask()命令所示;

[0055] 接着,当web服务器接收来自客户端的任务描述时,即建立web服务器与受控服务器之间的传输安全协议,如图3的步骤301:Create(SSH)(创建(SSH));

[0056] 受控服务器响应步骤301:Create(SSH)的任务执行,产生第一个输出日志,Log(syslog)321至日志服务器;

[0057] 并且,该受控服务器在与web服务器之间的传输安全协议,如SSH,建立之后,受控服务器会向web服务器要求取得任务描述的相关资讯,如图3的步骤302:GetTaskInfo()(获得任务信息),且在发出取得任务描述的请求后,即响应步骤302的执行而产生第二个输出日志,Log(syslog)322至日志服务器;

[0058] 接着,回应受控服务器的请求,web服务器将相关的任务描述转发给受控服务器,如图3的步骤303:TaskInfo(任务信息),且在取得任务描述后,即响应步骤303的执行而产生第三个输出日志,Log(syslog)323至日志服务器;

[0059] 一般来说,前述受控服务器由web服务器中取得的任务描述包括web服务器的IP地址以及描述任务的编号,其中所述地址与编号可以作为SSH的密钥认证,借由这些消息,受控服务器向web服务器发送获取脚本的请求,如图3的步骤304:GetScript()(获取脚本);

[0060] 然后,受控服务器可先自web服务器下载代理,并使代理根据描述任务的标号下载相对应的任务信息,如前置任务,脚本名称及脚本参数等,如图3的步骤305:script(脚本);

[0061] 在完成脚本的下载之后,受控服务器就可以直接执行对应任务标号的相关脚本,并将执行的结果输出以及回报执行结束给web服务器,如图3的步骤306:Msg*,步骤307:结束。当然受控服务器也会在执行完步骤306和307之后,响应产生最后二个输出日志,Log(syslog)324,325至日志服务器,以进行归类与储存。关于本发明的方法的其他细节已经参照图1进行了描述,在此不再累述。

[0062] 综上所述,本发明的分布式调度方法增强了服务器主机的安全、降低了运维工作的复杂度、实现了脚本的异步执行,从而降低了运维部署的成本。

[0063] 以上所述仅为本发明的实施例而已,并不用于限制本发明,对于本领域的技术人员来说,本发明可以有各种更改和变化。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的权利要求范围之内。

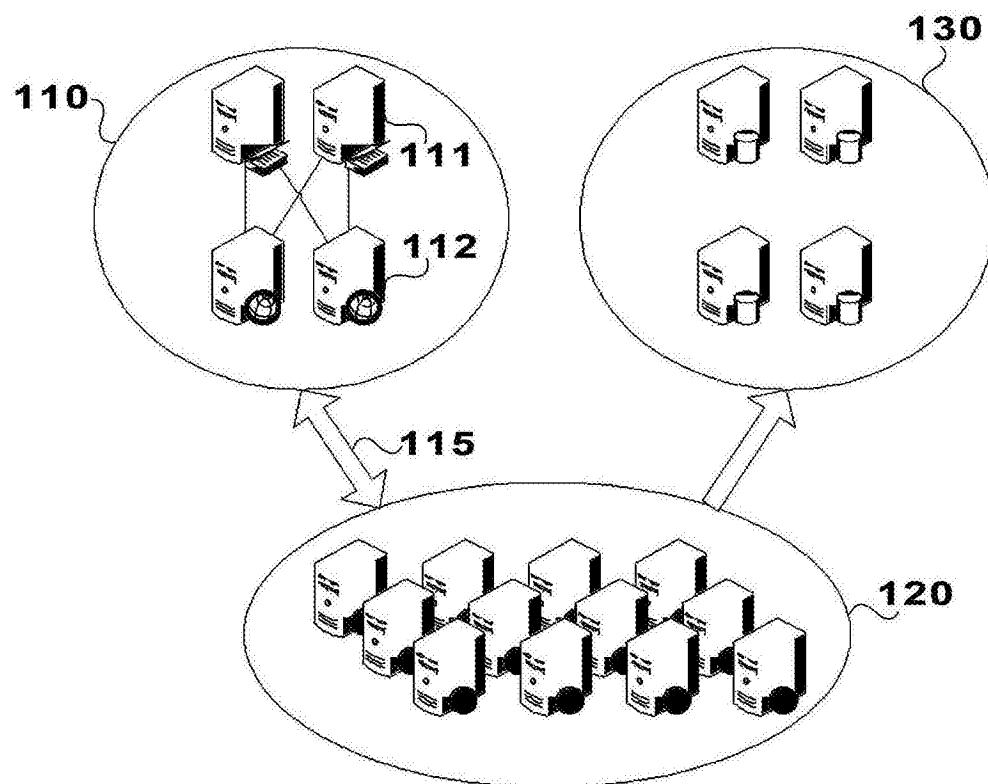


图1

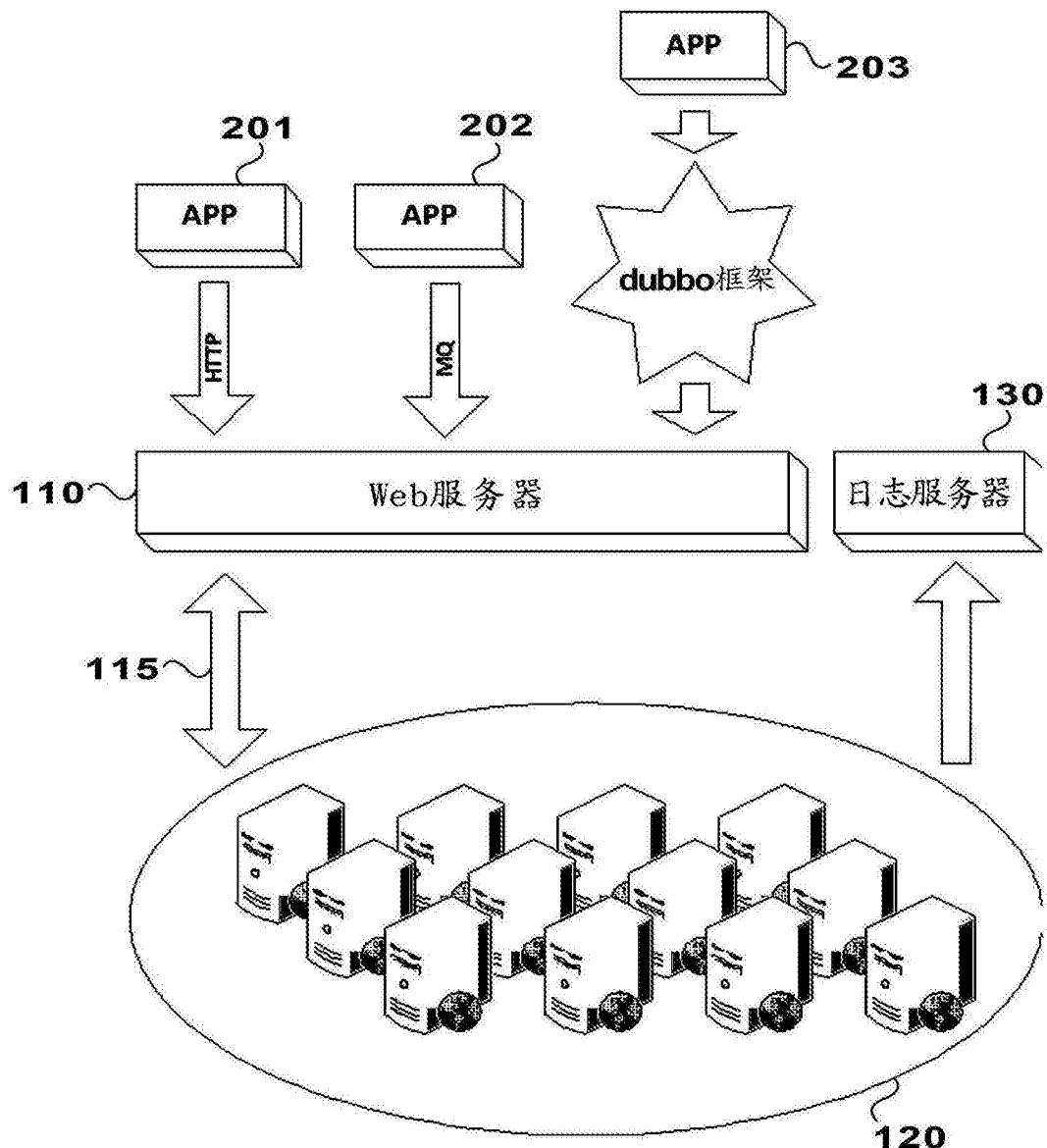


图2

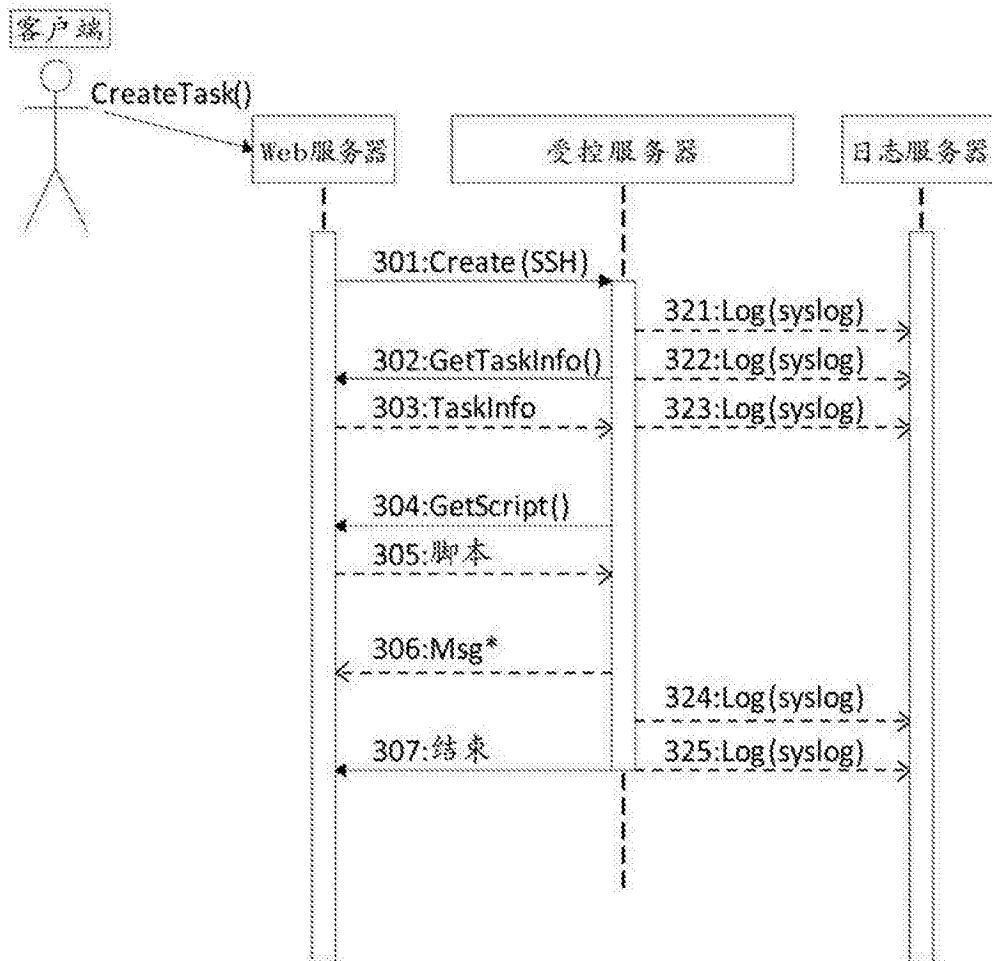


图3