



(12)发明专利

(10)授权公告号 CN 104205109 B

(45)授权公告日 2017.12.19

(21)申请号 201380017713.2

(22)申请日 2013.03.14

(65)同一申请的已公布的文献号
申请公布号 CN 104205109 A

(43)申请公布日 2014.12.10

(30)优先权数据
61/617474 2012.03.29 US
13/713536 2012.12.13 US

(85)PCT国际申请进入国家阶段日
2014.09.29

(86)PCT国际申请的申请数据
PCT/US2013/031115 2013.03.14

(87)PCT国际申请的公布数据
W02013/148223 EN 2013.10.03

(73)专利权人 微软技术许可有限责任公司
地址 美国华盛顿州

(72)发明人 B.A.伊阿卡斯 H.菲托西
B.施勒辛格 E.亚里夫

(74)专利代理机构 永新专利商标代理有限公司
72002
代理人 王英

(51)Int.Cl.
G06F 21/53(2006.01)

(56)对比文件
US 6463446 B1,2002.10.08,
US 7559065 B1,2009.07.07,
US 2011/0219385 A1,2011.09.08,
US 2010/0325414 A1,2010.12.23,

审查员 张莹

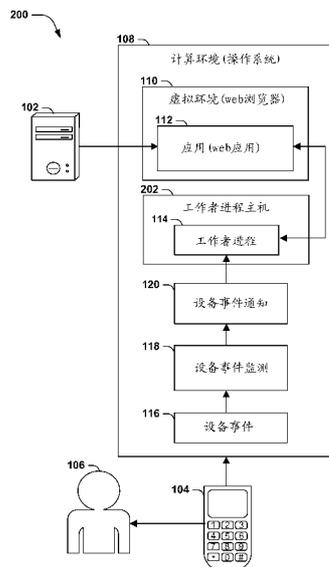
权利要求书2页 说明书13页 附图8页

(54)发明名称

持续性和弹性的工作者进程

(57)摘要

在计算领域内,许多情形涉及在虚拟环境内执行应用(例如执行在web浏览器内的web应用)。为了实施后台处理,这样的应用可以在虚拟环境内调用工作者进程;但是这种配置将工作者进程的生命周期与应用和/或虚拟环境的生命周期耦合。这里给出了用于在虚拟环境外部独立于应用的生命周期来执行工作者进程从而使得后台计算可以在应用和/或虚拟环境被终止之后并且甚至在计算环境重新启动之后持续进行的技术,以及用于在工作者进程实现执行事件时通知应用的技术(例如即使在应用未在执行时仍然检测设备事件)。这样的技术可以提高工作者进程的弹性和持续性,并且扩展执行在虚拟环境内的应用的能力。



1. 一种用于代表设备的应用来实施处理的系统,所述系统包括:
其中至少一项应用执行在设备上的虚拟环境;以及
在所述虚拟环境外部执行在所述设备上的工作者主机组件,其被配置成:
在从应用接收到针对执行涉及设备事件的工作者进程的请求时,发起在所述虚拟环境外部执行所述工作者进程;
在检测到所述应用终止时,避免终止所述工作者进程;以及
在所述工作者进程实现所述设备事件时,通过重启所述应用或者通过将设备事件通知排队直到稍后重新发起所述应用来向所述应用通知所述工作者进程的所述设备事件。
2. 根据权利要求1所述的系统:
所述虚拟环境包括web浏览器;
所述应用包括web应用;以及
所述工作者进程包括在所述web浏览器外部执行的web工作者。
3. 根据权利要求2所述的系统,所述工作者主机组件被配置成根据所述web浏览器的跨域限制策略来限制所述工作者进程的执行。
4. 根据权利要求2所述的系统,所述工作者主机组件包括被配置成将所述工作者进程作为本地web服务器进程来执行的本地web服务器。
5. 根据权利要求1所述的系统:
所述工作者主机组件包括在所述设备的计算环境内执行的后台服务;以及
所述工作者主机组件被配置成在所述设备重新启动所述计算环境时自动重新启动。
6. 根据权利要求5所述的系统,其还包括:工作者主机管理组件,其被配置成在检测到所述工作者主机组件的工作者主机组件故障时:
终止所述工作者主机组件;
在所述设备的所述计算环境内重新发起所述工作者主机组件的执行;以及
重新发起在所述工作者主机组件故障期间由所述工作者主机组件执行的工作者进程。
7. 根据权利要求5所述的系统:
所述工作者进程接收自服务器;以及
重新发起所述工作者进程的执行包括:
在重新启动所述计算环境之后,从所述服务器请求第二工作者进程;以及
在从所述服务器接收到所述第二工作者进程时,在所述虚拟环境外部发起所述第二工作者进程的执行。
8. 根据权利要求1所述的系统:
所述工作者进程接收自第一服务器;以及
在所述虚拟环境外部执行所述工作者进程包括:限制所述工作者进程与所述第一服务器之外的其他服务器之间的通信。
9. 根据权利要求1所述的系统,所述工作者主机组件还被配置成在终止所述虚拟环境时避免终止所述工作者进程。
10. 根据权利要求1所述的系统:
所述设备包括计算环境;
终止所述应用包括:终止所述应用的同时重新启动所述设备的所述计算环境;以及

所述工作者主机组件还被配置成在重新启动所述设备的所述计算环境之后,在所述虚拟环境外部重新发起所述工作者进程的执行。

持续性和弹性的工作者进程

背景技术

[0001] 在计算领域内,许多情形涉及设备的虚拟环境内的应用的执行,比如执行在web浏览器内的web应用;针对设备所提供的计算环境之外的其他计算环境所开发的代码(例如针对Java虚拟机设计的Java代码);以及执行在隔离虚拟机内的不受信任的代码。虚拟机内的应用的执行(而不是作为设备的本机进程)可以在执行期间促进应用的兼容性和安全性。

[0002] 这样的应用常常涉及后台处理,比如长期运行的计算密集的进程。举例来说,应用可以针对特定类型的事件(比如移动电话上的传入呼叫)对设备的硬件组件进行监测。但是如果这样的处理由所述应用实施,则该应用的其他责任可能被不合期望地延迟;例如计算密集的处理可能抢先应对图形用户接口(GUI)消息,并且可能导致所述应用呈现出交互性或性能降低。为了在减少应用的其他处理责任的中断的同时实现这样的后台处理,所述应用可以调用虚拟环境中的一个或更多工作者进程(worker process)。举例来说,在超文本标记语言(HTML)的最近版本中提供的“web工作者”模型允许应用请求web浏览器在所述web浏览器内发起单独的工作者进程,从而通过安全且高性能的方式代表所述应用实现后台处理。

发明内容

[0003] 提供该发明内容部分是为了以简化形式介绍将在后面的具体实施方式部分中进一步描述的概念的选择。该发明内容部分不意图标识出所要求保护的主体内容的关键因素或实质特征,也不意图被用来限制所要求保护的主体内容的范围。

[0004] 虽然代表执行在虚拟环境中的应用的工作者进程的实例化给出了一些优点,但是同样在虚拟环境中执行那些工作者进程的模型可能存在显著的缺点。作为第一示例,“web工作者”模型通常把web工作者的生命周期耦合到web应用的生命周期;也就是说当web应用被挂起或终止时(比如通过关闭托管所述web应用的页面),web工作者也被终止。工作者进程还可以在托管所述工作者进程的虚拟环境终止(例如关闭web浏览器)时被终止,和/或在重新启动设备的计算环境(例如挂起或重新引导设备的操作系统)时被终止。在某些情形中,将工作者进程的生命周期与应用和/或虚拟环境的生命周期相关联可能是合乎期望的。但是在其他情形中,可能希望允许工作者进程在这样的事件之后持续进行,和/或持续经过例如操作系统的重新引导之类的设备事件。

[0005] 这里给出了用于扩展在虚拟环境(比如web浏览器或虚拟机)内代表应用执行的工作者进程的弹性的技术。根据这样的技术,并非在虚拟环境内执行工作者进程,而是所述设备可以在虚拟环境外部执行工作者进程。当应用和/或虚拟环境被终止时,设备可以避免终止工作者进程,从而将工作者进程的生命周期与应用的生命周期分开。可选的是,工作者进程还可以被配置成持续经过在虚拟环境外部发生的潜在中断事件,比如托管工作者进程的进程中的故障或者操作系统重新引导。当工作者进程实现执行事件时(例如完成复杂的计算进程或者检测到特定类型的设备事件),可以对执行在虚拟环境内的应用进行通知(可选地,如果应用当前未在执行的话重新发起所述应用)以便履行对于由工作者进程所实现的

执行事件的应对。通过把工作进程的执行与虚拟环境内的应用的执行解除耦合,这里给出的技术可以扩展能够由虚拟环境内的应用所实现的后台处理的弹性和能力。

[0006] 为了实现前述和有关目的,后面的描述和附图阐述了特定的说明性方面和实现方式。这些仅仅表明可以据以采用一个或更多方面的各种方式当中的几种。通过在结合附图时考虑的后面的详细描述,本公开内容的其他方面、优点和新颖特征将变得显而易见。

附图说明

[0007] 图1是刻画了代表web浏览器内的web应用的工作进程的执行的示例性情形的图示。

[0008] 图2是刻画了根据这里给出的技术的代表应用的工作进程的执行的示例性情形的图示。

[0009] 图3是根据这里给出的技术的将设备配置成履行与本地事件有关的客户端应用的请求的示例性方法的图示。

[0010] 图4是图示出了根据这里给出的技术的用于履行与本地事件有关的客户端应用的请求的示例性系统的组件方框图。

[0011] 图5是包括被配置成具体实现这里所阐述的规定中的一项或更多项的处理器可执行指令的示例性计算机可读介质的图示。

[0012] 图6是刻画了根据这里给出的技术的执行在设备网格内的设备上的应用的工作进程的示例性情形的图示,其中工作进程可以尝试与所述应用的服务器、另一服务器以及设备网格中的另一设备的远程进程进行通信。

[0013] 图7是说明根据这里给出的技术的代表执行在虚拟环境内的应用的持续性工作进程的执行的示例性情形的图示。

[0014] 图8图示出了可以在其中实施这里所阐述的规定中的一项或更多项的示例性计算环境。

具体实施方式

[0015] 现在将参照附图来描述所要求保护的主体内容,其中相同的附图标记始终被用来指代相同的元件。在后面的描述中,出于解释的目的阐述了许多具体细节以便提供关于所要求保护的主体内容的透彻理解。但是可能明显的是,可以在没有这些具体细节的情况下实践所要求保护的主体内容。在其他实例中,以方框图形式示出了结构和设备以便于描述所要求保护的主体内容。

[0016] A、介绍

[0017] 在计算领域内,许多情形涉及在虚拟环境内在设备上执行应用,而不是在设备的本机计算环境内执行。作为第一示例,web浏览器可以提供用于执行独立于设备的代码(比如JavaScript和超文本标记语言(HTML)内容)的标准化平台。作为第二示例,Java虚拟机(JVM)可以提供通用计算设备与代表Java应用的设备的本机能力之间的桥梁(例如将通用存储器分配请求翻译成所述设备的存储器模型)。作为第三示例,通过在第二设备内提供对于第一设备的模拟的仿真器,被设计用于第一设备的本机环境的代码可以完全在第二设备上执行。作为第四示例,可以在“沙盒”或人工机器(artificial machine)内执行不受信任

的代码,从而可以将恶意操作隔离到所述人工机器,并且可以由设备评估不受信任的代码对人工设备的影响而不存在对设备破坏的风险。

[0018] 这些和其他情形给出在虚拟环境内执行应用。在许多这样的情形中,所述应用可能涉及将以高性能方式应对的一些前台责任;例如呈现图形用户接口 (GUI) 的应用可能被设计成以高响应度履行与视觉控制的交互,这是因为即使小的延迟对于用户而言都是明显的。这样的应用还可能涉及可以通过更加缓和的方式实施的大量后台处理,比如长期运行的计算、将在空闲时间期间实施的维护任务以及对于设备事件的监测。

[0019] 为了平衡对于前台处理和后台处理二者的应对,虚拟环境和应用可以提供例如抢先多任务处理之类的技术,其中应用可以为各个任务指定优先级。但是这样的技术可能不完全足以调度在应用中所涉及的多种处理任务。作为这方面的第一示例,抢先于某些类型的后台处理(例如涉及资源的排他性锁定的后台处理)以便应对前台处理可能是不可能的或者不是高效的。作为这方面的第二示例,如果后台处理涉及监测设备事件,则这样的后台处理的中断可能导致无法检测到这样的设备事件。

[0020] 鉴于这些困难,一些虚拟环境允许应用在所述虚拟环境内发起单独的工作者进程。通过把工作进程与应用的执行线程分开,虚拟环境可以利用设备的计算环境(例如由操作系统提供的更加精密的抢先调度器)和/或计算硬件的并行处理能力(例如在多处理器设备中的不同处理器上或者在多核处理器的不同核心上并行执行工作者进程和应用线程)。此外,在与应用相同的虚拟环境内执行工作者进程可以提供安全性(例如对于工作者进程实行与对应用相同的安全性策略)和/或效率(例如允许应用与工作者进程在虚拟环境内作为对等方的紧密耦合)方面的优点。

[0021] 图1给出了刻画了用户106的设备104的示例性情形100的图示,其给出包括虚拟环境110(例如web浏览器)的计算环境108(例如操作系统)。设备104可以与一台或更多台服务器102(例如web服务器)通信,并且可以取回将代表用户106执行的一项或更多项应用112(例如嵌入在由用户106请求的web页面中的web应用)。因此,设备104可以在虚拟环境110内发起应用112的执行。此外,虚拟环境110可以被配置成允许应用112请求执行工作者进程114,比如将代表应用112实施的后台维护任务,而无需中断应用112的前台处理的执行。因此,虚拟环境110可以与应用112并排地在虚拟环境110内发起工作者进程114的执行。举例来说,工作者进程114可以请求向其通知各种设备事件116,比如通过电话之类的移动通信设备接收到的传入呼叫或消息。设备104可以在计算环境108内应用某种设备事件监测118,并且在检测到设备事件116时可以向虚拟环境110内的工作者进程114递送设备事件通知120。通过这种方式,设备104可以允许应用112在虚拟环境110内调用工作者进程114以实现各种类型的后台处理,而无需中断应用112的前台处理。可以认识到,在图1的示例性情形100中图示出的模型准确地描述了存在于超文本标记语言 (HTML) 的近来版本中的“web工作者”模型,其中JavaScript脚本可以在与脚本并排地执行在web浏览器内的“web工作者”进程中调用后台处理。

[0022] 虽然在图1的示例性情形100中给出的架构可以给出一些优点,但是可以认识到,其中也可能出现一些缺点。具体来说,与应用112并排地在虚拟环境110内执行工作者进程114在某些方面可能是不利的。举例来说,在某些情形中,可能希望将工作者进程114的生命周期与应用112耦合,从而使得应用112的挂起或终止也导致工作者进程114的挂起或终止。

但是在其他情形中,可能希望将应用112与工作者进程114解除耦合。举例来说,工作者进程114可能包括用户106可能希望完成的长期运行的进程,但是用户106可能不希望保留虚拟环境110内的应用112(例如用户106可能有意地或无意地导览离开托管web应用的web页面,从而导致终止与所述web应用相关联的web工作者)。作为另一个示例,用户106可能希望工作者进程114无限期地持续,比如用于发生在设备104上的特定类型的设备事件116的处理程序(例如用以在接收到文字消息或传入电话呼叫时通知用户106的通知进程)。此外,在某些情形中,可能希望允许工作者进程114不仅在应用112的挂起或终止之后持续,而且还在虚拟环境110的挂起或终止之后持续(例如在用户106完全终止web浏览器之后),和/或在计算环境108的中断之后持续(例如在设备104重新引导或者以其他方式重新启动之后)。但是严格涉及工作者进程114的生命周期与应用112的生命周期的耦合的模型可能无法提供这样的持续的有弹性的工作者进程114。

[0023] B、所给出的技术

[0024] 这里给出了用于改进代表执行在虚拟环境内的应用112而执行的工作者进程114的持续性和/或弹性的技术。根据这些技术(并且与例如在图1的示例性情形100中所示出的那些技术之类的其他技术大不相同的是),并非在虚拟环境110内执行工作者进程114,而是设备104可以代表应用112但是在虚拟环境110之外执行工作者进程114。此外,当设备终止应用112时(例如暂时地或无限期地挂起应用112或者在完成或中止应用112的执行时),设备104可以避免终止与应用112相关联的一个或多个工作者进程114。通过这种方式,设备104可以允许执行持续的工作者进程114,该持续的工作者进程114即使在应用112当前未在执行时也继续服务于应用112的计算处理。

[0025] 图2给出了刻画这里给出的技术的示例性应用的示例性情形200的图示。在该示例性情形200中,设备104再次包括计算环境108,所述计算环境108包括可以在其中执行接收自服务器102的应用112的虚拟环境110(例如接收自web服务器并且在web浏览器内执行的web应用)。应用112可以请求某种后台处理,并且虚拟环境110可以通过代表应用112发起工作者进程114而遵从。但是根据这里给出的技术,可以由工作者进程主机202在虚拟环境110外部执行工作者进程114,所述工作者进程主机202可以独立于虚拟环境110对应用112的生命周期和资源请求的管理来管理工作者进程114的生命周期和资源请求。具体来说,在检测到应用112的终止时,设备工作者进程主机202可以避免终止工作者进程114;也就是说可以将工作者进程114的生命周期与应用112的生命周期分开并且独立管理。工作者进程主机202还可以允许工作者进程114在该工作者进程114实现执行事件时通知应用112。举例来说,由工作者进程114实施的后台处理可以涉及检测客户端设备事件116,比如来自移动电话的传入呼叫。设备可以利用某种设备事件监测118来代表工作者进程114检测设备事件116,并且可以在检测到这样的设备事件116时向工作者进程114递送设备事件通知120。特别地,即使应用112当前未在执行(并且可选地,即使虚拟环境110未在设备104上执行),该检测和递送也可以在工作者进程114操作的同时成功完成。此外,工作者进程主机202可以促进工作者进程114向应用112通知设备事件116(例如,如果应用112和/或虚拟环境110当前未在执行,则设备104可以重新发起应用112和/或虚拟环境110,或者可以简单地将设备事件通知120排队直到稍后由用户106重新发起这样的进程为止)。通过这种方式,设备104实现了独立于应用112的执行的执行的工作者进程114的持续执行。这些和其他优点可以是可通过

图2的示例性情形200中所图示的架构和这里给出的技术来实现的。

[0026] C、示例性实施例

[0027] 图3给出了这里给出的技术的第一示例性实施例,其被图示为代表执行在设备104的虚拟环境110内的应用112来实施处理的示例性方法300。示例性方法300例如可以被实施为存储在设备的存储器组件中的指令集,所述存储器组件比如是存储器电路、硬盘驱动器的盘片、固态存储设备或者磁盘或光盘,所述指令集被组织成使得当在设备的处理器上执行时,其使得所述设备根据这里给出的技术来操作。示例性方法300开始于302处,并且涉及在设备的处理器上执行304指令。具体来说,这些指令可以被配置成在从应用112接收到针对执行涉及执行事件的工作者进程114的请求时,在虚拟环境110外部发起306工作者进程114的执行。所述指令还被配置成在检测到应用112的终止时,避免308终止工作者进程114。所述指令还被配置成在工作者进程114实现所述执行事件时,向应用112通知310工作者进程114的执行事件。在代表应用112完成处理之后,示例性方法300根据这里给出的技术代表应用112给出工作者进程114的执行,并且因而结束于312处。

[0028] 图4给出了这里给出的技术的第二示例性实施例,其被图示为刻画了示例性系统408的示例性情形400,所述示例性系统408被配置成代表执行在设备104的虚拟环境110内的应用来实施处理。示例性系统408例如可以被实施为存储在设备402的存储器组件中的指令集,所述指令集被配置成当在设备402的处理器404上执行时,使得设备402根据这里给出的技术来操作。示例性系统408包括虚拟环境110,在该虚拟环境110中,至少一项应用112在设备104上执行。示例性系统408还包括在虚拟环境110外部在设备104上执行的工作者主机组件410,该工作者主机组件410被配置成在从应用112接收到针对执行涉及执行事件406的工作者进程114的请求时,在虚拟环境110外部发起工作者进程114的执行。工作者主机组件410还被配置成在检测到应用112的终止时,避免终止工作者进程114;并且在工作者进程114实现执行事件406时,向应用112通知工作者进程114的执行事件406。通过这种方式,示例性系统408根据这里给出的技术代表执行在设备402的虚拟环境110内的应用112实现工作者进程114的执行。

[0029] 再一个实施例涉及包括处理器可执行指令的计算机可读介质,所述处理器可执行指令被配置成应用这里给出的技术。这样的计算机可读介质例如可以包括涉及有形设备的计算机可读存储介质,比如存储器半导体(例如利用静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)和/或同步动态随机存取存储器(SDRAM)技术的半导体)、硬盘驱动器的盘片、闪存设备或者磁盘或光盘(比如CD-R、DVD-R或软盘),其编码计算机可读指令集,当由设备的处理器执行时,所述计算机可读指令集使得所述设备实施这里给出的技术。这样的计算机可读介质还可以包括(作为与计算机可读存储介质不同的一类技术的)各种类型的通信介质,比如可以通过各种物理现象传播的信号(例如电磁信号、声波信号或光学信号)以及在各种有线情形(例如通过以太网或光纤线缆)和/或无线情形(例如,诸如WiFi之类的无线局域网(WLAN)、诸如蓝牙之类的个域网(PAN)或者蜂窝或无线网络)中传播的信号,其编码计算机可读指令集,当由设备的处理器执行时,所述计算机可读指令集使得所述设备实施这里给出的技术。

[0030] 在图5中图示出了可以通过这些方式设想到的示例性计算机可读介质,其中实现方式500包括计算机可读介质502(例如CD-R、DVD-R或硬盘驱动器的盘片),其上编码计算机

可读数据504。该计算机可读数据504又包括被配置成根据这里阐述的原理操作的计算机指令集506。在这方面的一个实施例中,处理器可执行指令506可以被配置成实施用于履行客户端应用108的与本地事件105有关的请求的方法508,比如图3的示例性方法300。在这方面的另一个实施例中,处理器可执行指令506可以被配置成实施用于履行客户端应用108的与本地事件206有关的请求的系统,比如图4的示例性系统408。该计算机可读介质的一些实施例可以包括计算机可读存储介质(例如硬盘驱动器、光盘或闪存设备),所述计算机可读存储介质被配置成存储通过这种方式配置的处理器可执行指令。本领域普通技术人员可以设想到被配置成根据这里给出的技术操作的许多这样的计算机可读介质。

[0031] D、变型

[0032] 在许多方面可以设想到这里讨论的技术的变型,并且一些变型可以相对于这些和其他技术的其他变型给出附加的优点和/或减少缺点。此外,一些变型可以被组合实施,并且通过协同合作,一些组合可以刻画附加的优点和/或减少的缺点。所述变型可以被合并并在各个实施例中(例如图3的示例性方法300和图4的示例性系统408),以便依据这样的实施例给予单独的和/或协作的优点。

[0033] D1、情形

[0034] 在这些技术的实施例之间可以有所不同的第一方面涉及可以在其中利用这样的技术的情形。

[0035] 作为该第一方面的第一变型,这里给出的技术可以与许多类型的设备402(比如服务器、服务器群、工作站、膝上型设备、平板设备、移动电话、游戏控制台以及网络器具)一起使用。这样的设备402还可以提供多种计算组件,比如有线或无线通信设备;人类输入设备,比如键盘、鼠标、触摸板、触敏显示器、麦克风以及基于手势的输入组件;自动化输入设备,比如静止或运动摄影机、全球定位服务(GPS)设备以及其他传感器;比如显示器和扬声器之类的输出设备;以及通信设备,比如有线和/或无线网络组件。

[0036] 作为该第一方面的第二变型,这里给出的技术可以与各种类型的服务器102(比如web服务器、文件服务器、应用服务器、媒体服务器、对等共享协调服务器、数据库服务器、电子邮件服务器、物理量度监测服务器以及监控和数据采集(SCADA)自动化服务器)一起使用。

[0037] 作为该第一方面的第三变型,这里给出的技术可以被利用来服务于执行在许多类型的虚拟环境110内的应用112,所述虚拟环境110比如是web浏览器、模拟和/或仿真虚拟机、中间件平台和/或隔离构造(isolation construct)。

[0038] 作为该第一方面的第四变型,这里给出的技术可以被利用来服务于许多类型的应用112,比如用各种解译和/或编译语言编写的web应用、文件共享应用、媒体渲染应用以及数据驱动的客户应用。

[0039] 作为该第一方面的第五变型,这里给出的技术可以涉及在许多类型的工作者进程主机202内执行许多类型的工作者进程114。举例来说,工作者进程114可以包括根据HTML 5“web工作者”模型执行但是在托管应用112的web浏览器外部的单独外壳中执行的JavaScript脚本。可替换地,工作者进程114可以包括执行在设备104内的受管理情境(其中管理运行时间包括工作者进程主机202)中的部分或完全编译的本机执行的应用。作为另一种替换方案,工作者进程主机202可以包括被配置成代表执行在相同设备上的web浏览器内

的应用112来实施持续性后台处理的本地web服务器。此外,工作者进程114可以包括利用Node.js服务器侧JavaScript环境的JavaScript脚本。

[0040] 作为该第一方面的第六变型,工作者进程114可以代表应用112实施许多类型的后台处理,包括长期运行的计算进程(其中执行事件包括计算进程的完成);调度或定时器进程(其中执行事件包括定时器的超时或消逝);持续地可用的功能,比如守护(daemon)进程(其中执行事件包括由执行在设备104上的另一进程调用该功能);和/或监测各种类型的设备事件116(其中执行事件包括所检测到的设备事件116的实例,比如检测来自设备104的用户106的用户输入,或者在计算环境108内发生的硬件或软件事件,比如接收到移动电话设备的传入呼叫或消息)。工作者进程114可以对这样的设备事件116应用许多类型的处理,比如向应用112报告设备事件116,将设备事件116的发生记入日志,或者在设备事件116的应对期间应用附加的逻辑。这些和其他情形可以与这里给出的技术兼容并且可以有利地利用这里给出的技术。

[0041] D2、工作者进程托管

[0042] 在这里给出的技术的实施例之间可以有所不同的第二方面涉及在虚拟环境110外部托管工作者进程114的执行的方式。也就是说,在工作者进程114从应用112接收到并且在虚拟环境110的外部执行时,可能不利和/或困难的是在没有某种类型的进程主机的情况下允许工作者进程114在虚拟环境110外部执行,因为这样可能导致将任意的不受信任的代码作为本机进程在设备104的计算环境108内执行。相应地,虽然与虚拟环境110分开,但是工作者进程114的执行可以由为工作者进程114提供各种类型的托管服务的工作者主机组件410照管。

[0043] 作为该第二方面的第一变型,工作者主机组件410可以对工作者进程114应用许多形式的托管。作为第一示例,工作者主机组件410还可以为工作者进程114提供虚拟环境110,只要为工作者进程114提供的虚拟环境110与为应用112提供的虚拟环境110分开。实际上,为工作者进程114提供的虚拟环境110可以相当类似于为应用112提供的虚拟环境110(例如全部两个进程都执行在web浏览器的情境内),并且虚拟环境110的分开允许这里所提供的工作者进程114独立于应用112的生命周期而持续。此外,为应用112提供的第一虚拟环境110与为工作者进程114提供的第二(分开的)虚拟环境110之间的相似性可以实现一些便利和高效;例如对应用112应用的相同安全性策略可以被应用于工作者进程114。举例来说,在web浏览器内执行的web应用常常根据跨域安全性策略而受到限制,从而例如允许web应用只通过因特网与从中接收到所述web应用的域进行通信。工作者主机组件410可以根据被应用于执行在web浏览器内的web应用并且与工作者进程114相关联的相同跨域安全性策略来限制工作者进程114的执行,例如限制工作者进程114使其无法与为设备104提供应用112的域之外的任何域、服务器或设备进行通信。

[0044] 可替换地,工作者主机组件410可以为工作者进程114给出具有不同细节的不同执行模型。这样的不同之处可以鉴于应用112(例如适合于呈现用户接口的自组织前端代码的执行环境)和工作者进程114(例如适合于没有用户接口的情况下的计算复杂的长期运行的处理的执行环境)的不同情境来选择和/或设计。作为这方面的第一示例,由于工作者进程114可以利用比应用112更加计算复杂的进程,因此工作者主机组件410可以在执行之前或期间部分地和/或完全地编译工作者进程114,从而通过使用代码优化技术提供更加高性能

的执行,而应用112则可以作为可解译脚本来执行。作为这方面的第二示例,工作者主机组件410可以按照不同于在虚拟环境110内调度应用112的方式在计算环境108内调度工作者进程114的执行。举例来说,工作者进程114可以被调度为特别适合于在设备104的空闲时段期间的长时间相对不受中断的处理的长期运行的进程,而应用112则可以被调度为高度响应但是短暂的代码调用集合,比如对于用户接口事件的响应。作为这方面的第三示例,可以为工作者进程114提供不同于应用112的应用编程接口(API)集合,例如对应于工作者进程114的较低层级硬件和网络通信模块以及对应于应用112的用户接口类型模块。作为这方面的第四示例,工作者主机组件410可以给出不同于应用112的虚拟环境110的安全性模型(例如鉴于在虚拟环境110外部执行的工作者进程114的扩展的能力,安全性进程工作者主机组件410可以在执行之前和/或期间应用更加严格的代码评估以便检测和阻断恶意操作)。相反,在其他方面,工作者进程114的安全性模型可以比用于执行在虚拟环境110内的应用112的安全性模型更加宽松;举例来说,由于可以通过通信等待时间方面的可接受的增加而更加密切地详审工作者进程114与其他设备之间的通信,因此可以不根据对应用112实行的跨域安全性策略来限制工作者进程114。举例来说,如果设备104可以访问执行在第二设备上的远程进程,则工作者主机组件410可以在接收到针对把执行在第二设备上的远程进程与工作者进程相连的请求时将所述远程进程与工作者进程114相连。功能的扩展可以反映出工作者进程114中的比应用112更高的信任水平,并且可以视用户同意而定;例如工作者主机组件410可以向用户106给出在虚拟环境110外部发起工作者进程114的执行的提议,并且可以只有在从用户106接收到对于所述提议的接受的情况下才发起这样的执行。还可以允许用户106检查在相关联的应用112终止之后持续进行的工作者进程114,以及改动其执行(例如履行用户106针对挂起和/或终止工作者进程114的请求)。

[0045] 通过这些和其他方式,为工作者进程114提供的执行模型可以显著不同于为应用112所提供的执行模型。作为这方面的第一示例,工作者主机组件410可以包括允许工作者进程114的受管理执行的管理运行时间(例如根据设备104的架构促进存储器分配和组件访问请求)。作为这方面的第二示例,工作者主机组件410可以包括布置在设备104上的本地web服务器,其为同样在设备的web浏览器内执行的web应用服务,并且被配置成将工作者进程114作为本地web服务器进程来执行。举例来说,工作者主机组件410可以被设计成以本地web服务器的Node.js服务器侧脚本化模块为目标的JavaScript脚本。作为这方面的第三示例,工作者主机组件410可以提供与虚拟环境110分开执行的虚拟机(比如Java虚拟机),以便将工作者进程114的生命周期与应用112的生命周期分开。对于这里给出的技术,在由工作者主机组件410为工作者进程114提供的执行环境方面的许多选择可能是可行的,并且与之兼容。

[0046] 图6给出了刻画在由这里所提供的工作者主机组件410托管工作者进程114方面的几种变型的示例性情形600的图示。在该示例性情形600中并且根据这里给出的技术,设备104提供执行应用112的虚拟环境110,以及代表应用112但是在虚拟环境110外部执行工作者进程114的工作者主机组件410。作为这方面的第一实示例,在一些方面中,对于工作者进程114的托管可以与在虚拟环境110内托管应用112分享相似性,比如类似的跨域限制策略。举例来说,如果首先从提供第一服务(例如web应用服务)的服务器102接收到应用112,并且工作者进程114后来发起与相同的服务器102的通信,则工作者主机组件410可以允许与提

供第一服务的服务器102进行通信610。但是如果工作者进程114尝试发起与提供第一服务的服务器102之外的其他服务器614所提供的服务的通信612,则工作者主机组件410可以阻断616通信612。可替换地或附加地,被应用于工作者进程114的安全性策略在相同的或其他方面可以不同于对于应用112所实行的安全性策略。举例来说,在该示例性情形600中,设备104作为设备网格602的一部分操作,所述设备网格602包括由相同的用户106通过互操作方式来操作的设备104的集合,例如其中存在大量数据共享以便跨多个设备104为用户106提供一致的并且无缝的用户体验。相应地,即使应用112不被允许与提供该应用112的服务器102外部的远程进程606进行通信,工作者进程114也可以被允许与执行在设备网格602内的第二设备604上的远程进程606进行通信608。

[0047] D3、工作者进程的持续性和弹性

[0048] 在这些技术的实施例之间可以有所不同的第三方面涉及工作者进程114的持续性。除了避免在检测到应用112终止时终止工作者进程114的执行之外,这里给出的技术还可以利用附加的技术来促进工作者进程114的持续性和弹性。

[0049] 作为该第三方面的第一变型,即使不仅应用112终止而且虚拟环境110也终止,工作者主机组件410也可以允许工作者进程114持续进行(例如不仅在用户106关闭托管应用112的web页面之后持续进行,而且还在web浏览器被完全终止并且从存储器中卸载之后持续进行)。因此,即使在终止虚拟环境110时,设备104仍然可以避免终止工作者进程114。

[0050] 作为该第三方面的第二变型,即使在工作者主机组件410中发生故障(例如管理工作进程114的组件的异常、崩溃或终止),工作者主机组件410仍然可以允许工作者进程114持续进行。举例来说,工作者主机组件410可以周期性地记录执行中的工作者进程114及其来源的列表。如果在工作者主机组件410内发生故障,则设备104可以终止工作者主机组件410,在设备的计算环境内重新发起工作者主机组件的执行,并且重新发起在故障发生时执行在工作者主机组件410内的工作者进程114的执行。

[0051] 作为该第三方面的第三变型,即使在设备104的计算环境108中发生重大改变,工作者主机组件410甚至仍然可以允许工作者进程114持续进行。举例来说,在设备104的计算环境108重新启动(比如重新引导)期间可能发生应用112和/或虚拟环境110的终止。相应地,在重新启动计算环境108之后,设备可以在虚拟环境110外部重新发起工作者进程114的执行。作为这方面的第一示例,设备104可以存储工作者进程114并且可以周期性地记录其状态,从而在重新启动计算环境108之后,设备104可以利用相对新近的状态重新发起工作者进程114的执行。作为这方面的第二示例,如果工作者进程114从服务接收到和/或保持与之通信,则设备104可以在重新启动计算环境108之后从所述服务请求第二工作者进程114(例如被配置成根据第一工作者进程114的状态的近来观察或报告恢复执行的工作者进程114),并且可以在接收到来自所述服务的第二工作者进程114时在设备104的虚拟环境110外部执行第二工作者进程114。这样的持续性甚至可以允许工作者进程114在不同的设备104上重新启动或转移计算环境108时持续进行,例如在把工作者进程114转移到用户106的设备网格602内的第二设备104时持续进行。

[0052] 作为该第三方面的第四变型,可以与应用112共享工作者进程114的弹性。举例来说,除了在终止应用112时避免终止工作者进程114之外,工作者主机组件410还可以在检测到在虚拟环境110内重新发起应用112的执行时识别出由该应用112在终止之前发起的任何

当前正在执行的工作者进程114,并且可以将应用112与所识别出的工作者进程114重新连接。工作者进程114例如可以向应用112通知在该应用112的终止时段期间发生在设备104上的任何相关的设备事件116。用于扩展工作者进程114的持续性和弹性的这些和其他技术可以在这里给出的技术的实施期间设想到,并且可以与之兼容。

[0053] D4、应用通知

[0054] 在这些技术的实施例之间可以有所不同的第四方面涉及在相关联的工作者进程114实现执行事件406时通知应用112的方式。

[0055] 作为该第四方面的第一变型,可以通过许多类型的通知机制来实现对于应用112的通知。作为这方面的第一示例,工作者主机组件410可以简单地把执行事件406记入日志,并且应用112可以周期性地检查关于工作者进程114所实现的执行事件406的日志。作为这方面的第二示例,可以通过消息传递来实现通知;例如工作者进程114可以生成标识并描述执行事件406的执行事件通知412,并且虚拟环境110可以向应用112给出执行事件通知412。作为这方面的第三示例,可以通过回调(callback)机制实现通知,其中应用112表明在工作者进程114实现执行事件406时将要调用的代码的存储器位置。该代码例如可以包括将在虚拟环境110内执行的应用112的一部分;将由工作者主机组件410执行的工作者进程114的一部分;计算环境108的函数,比如API调用;或者单独的代码部分,比如移动代理、抽象句法树或者脚本。

[0056] 作为该第四方面的第二变型,可以在通知期间将工作者进程114的弹性扩展到应用112。作为这方面的第一示例,如果工作者进程114在应用112被终止之后实现了执行事件406,则设备104可以作为通知进程的一部分在虚拟环境110中重新发起应用112,这例如是通过重新加载创建该web工作者的web应用的web页面。作为这方面的第二示例,可以作为通知进程的一部分向用户106自动呈现应用112。举例来说,在实现执行事件406时,设备104可以将虚拟环境110和/或应用112提升到计算环境108内的前台呈现(例如把web浏览器带到前台并且切换到包含应用112的标签)。

[0057] 图7给出刻画了执行在设备104的计算环境108内的持续性和弹性的工作者进程的示例性情形的图示。在该示例性情形中,在第一时间点700处,web浏览器702(包括计算环境108内的虚拟环境110)可以从服务器102接收应用112,并且可以在该web浏览器702内发起应用112的执行。这一执行可以涉及用于提供某种后台处理的工作者进程114,其可以由计算环境108作为本地web服务器704的服务器进程来执行。此外,本地web服务器704可以记录工作者进程114,包括工作者进程114和应用112的来源的服务器102,以及周期性地记录工作者进程114的状态。

[0058] 在第二时间点706处,可以由用户106和/或设备104发起web浏览器702的终止708,并且这样的终止708可以包括终止执行在web浏览器702内的应用112。但是根据这里给出的技术,通过在web浏览器702外部执行工作者进程114,即使应用112和虚拟环境110终止708,也允许工作者进程114持续进行。

[0059] 在第三时间点710处,计算环境108也可能经历终止712(例如在设备104的重新引导期间)。在终止712之后的第四时间点714处,计算环境108可以被重新启动。所述重新启动可以涉及本地web服务器704的自动重新发起,其可以检查在计算环境108终止712时由本地web服务器704托管的工作者进程114的记录。本地web服务器704随后可以联系服务器102,

并且可以从服务器102接收716第二工作者进程114(例如与第一时间点700处发起的工作者进程114完全相同的实例,或者反映出在终止712时的工作者进程114的状态的经过改动的工作者进程)。因此计算环境108可以在该计算环境108重新启动之后自动重新发起工作者进程114的执行。

[0060] 在第四时间点718处,可能在计算环境108内发生设备事件116(例如针对用户106的传入呼叫)。如果工作者进程114与设备事件116相关联(例如作为工作者进程114的执行事件406),则本地web服务器704可以向工作者进程114通知设备事件116。此外,工作者进程114可以向应用112通知720设备事件116。附加地,如果应用112和/或web浏览器702当前未在执行,则设备104可以重新发起web浏览器702和/或应用112(例如通过在web浏览器702内重新加载所述页面),并且附加地,可以把web浏览器702带到计算环境108内的前台呈现。按照这种方式,通过这里给出的技术的实行,设备104可以实现弹性执行的工作者进程114,其即使在应用112终止时、在虚拟环境110终止时并且甚至在计算环境108重新启动时仍然持续进行。

[0061] E、计算环境

[0062] 图8和后面的讨论提供了用以实施这里所阐述的规定中的一项或更多项的实施例的适当计算环境的简要的一般性描述。图8的操作环境仅仅是适当的操作环境的一个示例,而不意图暗示关于操作环境的使用和功能的范围的任何限制。示例性的计算设备包括但不限于个人计算机、服务器计算机、手持式或膝上型设备、移动设备(比如移动电话、个人数字助理(PDA)、媒体播放器等等)、多处理器系统、消费电子装置、小型计算机、大型计算机、包括任何前述系统或设备的分布式计算环境等等。

[0063] 虽然并无要求,但是在由一个或更多计算设备执行的“计算机可读指令”的一般情境中描述了实施例。计算机可读指令可以经由计算机可读介质(后面讨论)来分发。计算机可读指令可以被实施为实施特定任务或者实施特定抽象数据类型的程序模块,比如函数、对象、应用编程接口(API)、数据结构等等。通常来说,在各种环境中可以按照希望组合或分配计算机可读指令的功能。

[0064] 图8图示出了包括计算设备802的系统800的示例,所述计算设备802被配置成实施这里提供的一个或更多实施例。在一种配置中,计算设备802包括至少一个处理单元806和存储器808。取决于计算设备的确切配置和类型,存储器808可以是易失性的(例如比如RAM)、非易失性的(例如比如ROM、闪存等等)或者二者的某种组合。这种配置在图8中由虚线804图示出。

[0065] 在其他实施例中,设备802可以包括附加的特征和/或功能。举例来说,设备802还可以包括附加的存储装置(例如可移除和/或不可移除的),其中包括但不限于磁性存储装置、光学存储装置等等。这样的附加存储装置在图8中由存储装置810图示出。在一个实施例中,用以实施这里提供的一个或更多实施例的计算机可读指令可以处于存储装置810中。存储装置810还可以存储用以实施操作系统、应用程序等等的其他计算机可读指令。计算机可读指令可以被加载到存储器808中以供例如处理单元806执行。

[0066] 这里使用的术语“计算机可读介质”包括计算机存储介质。计算机存储介质包括通过用于存储例如计算机可读指令或其他数据之类的信息的任何方法或技术实施的易失性和非易失性、可移除和不可移除的介质。存储器808和存储装置810是计算机存储介质的示

例。计算机存储介质包括但不限于RAM、ROM、EEPROM、闪存或其他存储器技术,CD-ROM、数字通用盘(DVD)或其他光学存储装置,磁盒、磁带、磁盘存储装置或其他磁性存储设备,或者可以被用来存储所期望的信息并且可以由设备802访问的任何其他介质。任何这样的计算机存储介质都可以是设备802的一部分。

[0067] 设备802还可以包括允许设备802与其他设备通信的(多个)通信连接816。(多个)通信连接816可以包括但不限于调制解调器、网络接口卡(NIC)、集成网络接口、射频传送器/接收器、红外端口、USB连接或者用于将计算设备802连接到其他计算设备或其他接口。(多个)通信连接816可以包括有线连接或无线连接。(多个)通信连接816可以传送和/或接收通信介质。

[0068] 术语“计算机可读介质”可以包括通信介质。通信介质通常在例如载波或其他传输机制的“已调数据信号”中具体实现计算机可读指令或其他数据,并且包括任何信息递送介质。术语“已调数据信号”可以包括其一项或更多项特性被设定或修改成在其中编码信息的信号。

[0069] 设备802可以包括(多个)输入设备814,比如键盘、鼠标、笔、语音输入设备、触摸输入设备、红外摄影机、视频输入设备和/或任何其他输入设备。在设备802中还可以包括(多个)输出设备812,比如一个或更多显示器、扬声器、打印机和/或任何其他输出设备。(多个)输入设备814和(多个)输出设备812可以经由有线连接、无线连接或其任意组合连接到设备802。在一个实施例中,来自另一个计算设备的输入设备或输出设备可以被用作计算设备802的(多个)输入设备814或(多个)输出设备812。

[0070] 计算设备802的组件可以通过各种互连(比如总线)相连接。这样的互连可以包括外围组件互连(PCI)(比如PCI Express)、通用串行总线(USB)、Firewire(IEEE 1394)、光学总线结构等等。在另一个实施例中,计算设备802的组件可以通过网络互连。举例来说,存储器808可以包括位于通过网络互连的不同物理位置的多个物理存储器单元。

[0071] 本领域技术人员将认识到,被利用来存储计算机可读指令的存储设备可以分布在网络上。举例来说,可经由网络818访问的计算设备820可以存储用以实施这里所提供的的一个或更多实施例的计算机可读指令。计算设备802可以访问计算设备820,并且下载所述计算机可读指令的一部分或全部以供执行。可替换地,计算设备802可以下载所需要的计算机可读指令的片段,或者一些指令可以在计算设备802处执行,并且一些指令可以在计算设备820处执行。

[0072] F、术语的使用

[0073] 虽然用特定于结构特征和/或方法动作的语言描述了主题内容,但是应当理解的是,在所附权利要求中限定的主题内容不一定受限于前面描述的具体特征或动作。相反,前面描述的具体特征和动作是作为实施权利要求的示例性形式而公开的。

[0074] 在本申请中使用的术语“组件”、“模块”、“系统”、“接口”等等通常意图指代与计算机有关的实体,或者硬件、硬件与软件的组合、软件或者执行中的软件。举例来说,组件可以但不限于运行在处理器上的进程、处理器、对象、可执行程序、执行线程、程序和/或计算机。作为说明,运行在控制器上的应用和控制器二者都可以是组件。一个或更多组件可以驻留在进程和/或执行的线程内,并且组件可以位于一台计算机上和/或分布在两台或更多台计算机之间。

[0075] 此外,所要求保护的主体内容可以被实施为利用标准编程和/或工程技术来产生软件、固件、硬件或其任意组合以便控制计算机实施所公开的主题内容的方法、设备、或者制品。这里使用的术语“制品”意图包含可以从任何计算机可读设备、载体或介质访问的计算机程序。当然,本领域技术人员将认识到,在不背离所要求保护的主体内容的范围或精神的情况下,可以对这种配置做出许多修改。

[0076] 这里提供了实施例的各个操作。在一个实施例中,所描述的操作中的一项或更多项可以构成存储在一个或更多计算机可读介质上的计算机可读指令,在由计算设备执行时,所述计算机可读指令将使得所述计算设备实施所描述的操作。据以描述所述操作中的一些或全部的顺序不应被解释为暗示着这些操作一定依赖于顺序。受益于本说明书,本领域技术人员将认识到替换的排序。此外,应当理解的是,并非所有操作都必然存在于这里提供的每一个实施例中。

[0077] 此外,这里使用的“示例性”一词意为充当示例、示例或说明。在这里被描述为“示例性”的任何方面或设计不一定被解释为比其他方面或设计更有利。相反,使用“示例性”一词是意图以具体的方式呈现概念。本申请中所使用的术语“或者”意图意味着包含性的“或者”而不是互斥性的“或者”。也就是说,除非另行指明或者从上下文明显看出,否则“X采用A或B”意图意味着自然的包含性排列中的任何一个。也就是说,如果X采用A;X采用B;或者X采用A和B二者,则“X采用A或B”在任何前述实例下都被满足。此外,除非另行指明或者从上下文中明显看出是针对单数形式,否则用在本申请和所附权利要求书中的冠词“一个”和“某一”通常可以被解释为意味着“一个或更多个”。

[0078] 此外,虽然关于一种或更多种实现方式示出并描述了本公开内容,但是基于阅读并理解本说明书和附图,本领域其他技术人员将会想到等效的更改和修改。本公开内容包括所有这样的修改和更改,并且仅由后面的权利要求书的范围来限制。特别关于由前面描述的组件(例如元件、资源等等)所实施的各项功能,被用来描述这样的组件的术语意图对应于(除非另行表明)实施所描述组件的指定功能的任何组件(例如其在功能上等效),尽管其在结构上并不等效于在本公开内容的这里说明的示例性实现方式中实施所述功能的所公开的结构。此外,虽然本公开内容的特定特征可能是关于几种实现方式当中的仅仅一种公开的,但是这样的特征可以与其他实现方式的一项或更多项其他特征组合,这可能对于任何给定或特定应用而言是所期望并且有利的。此外,就在具体实施方式部分或权利要求书中使用术语“包括”、“具有”、“含有”、“带有”或其变型而言,这样的术语意图按照与术语“包括”类似的方式进行包含。

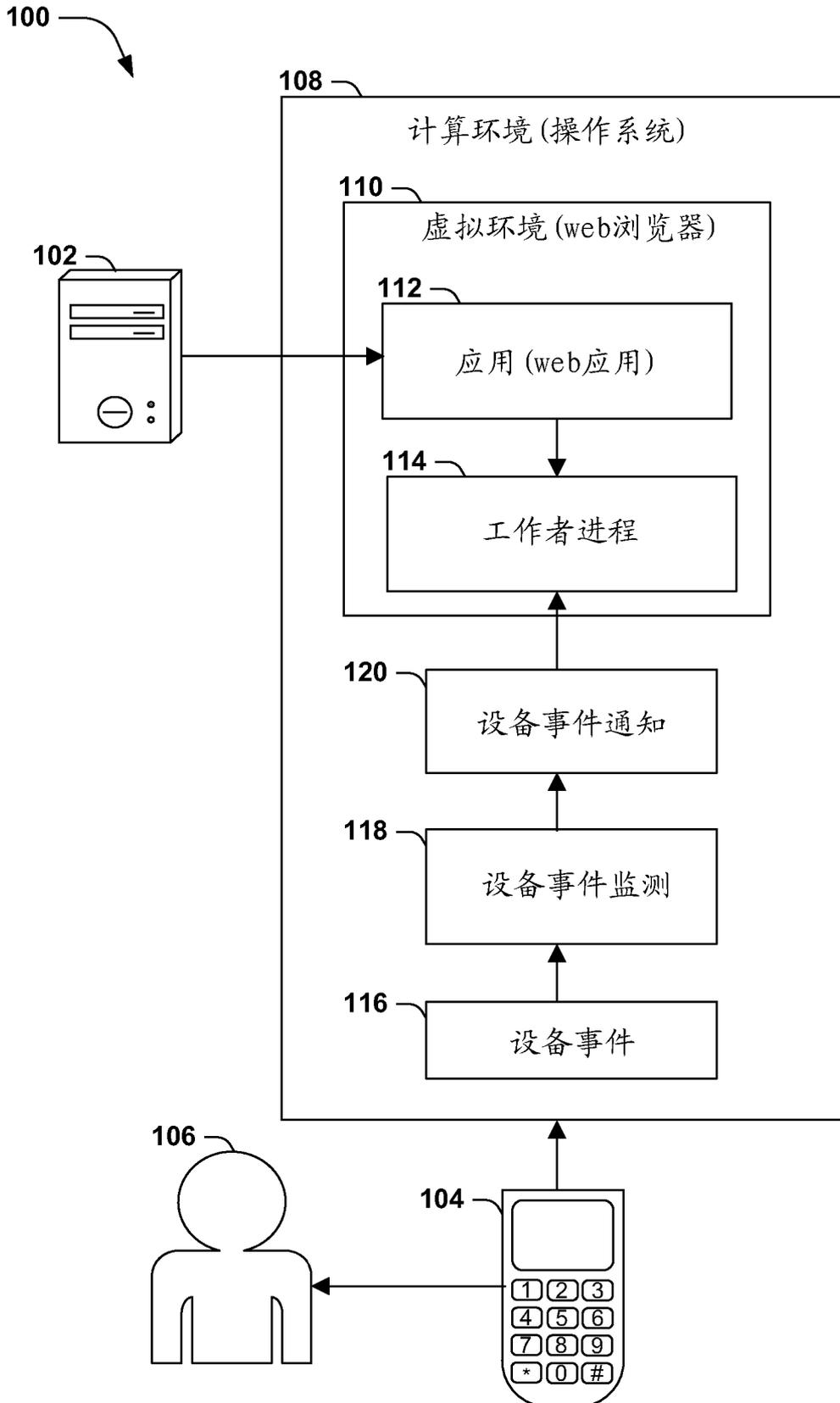


图 1

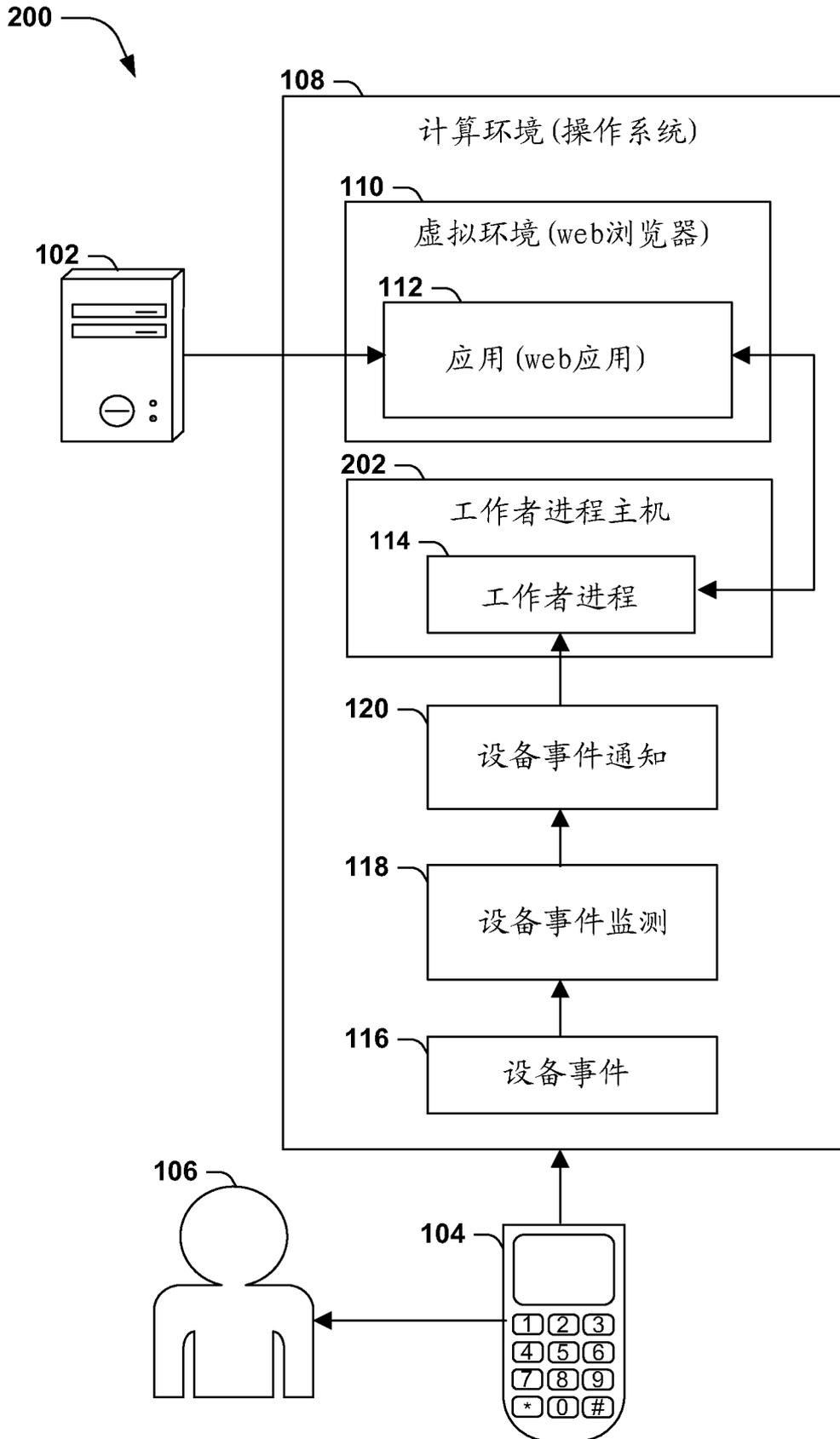


图 2

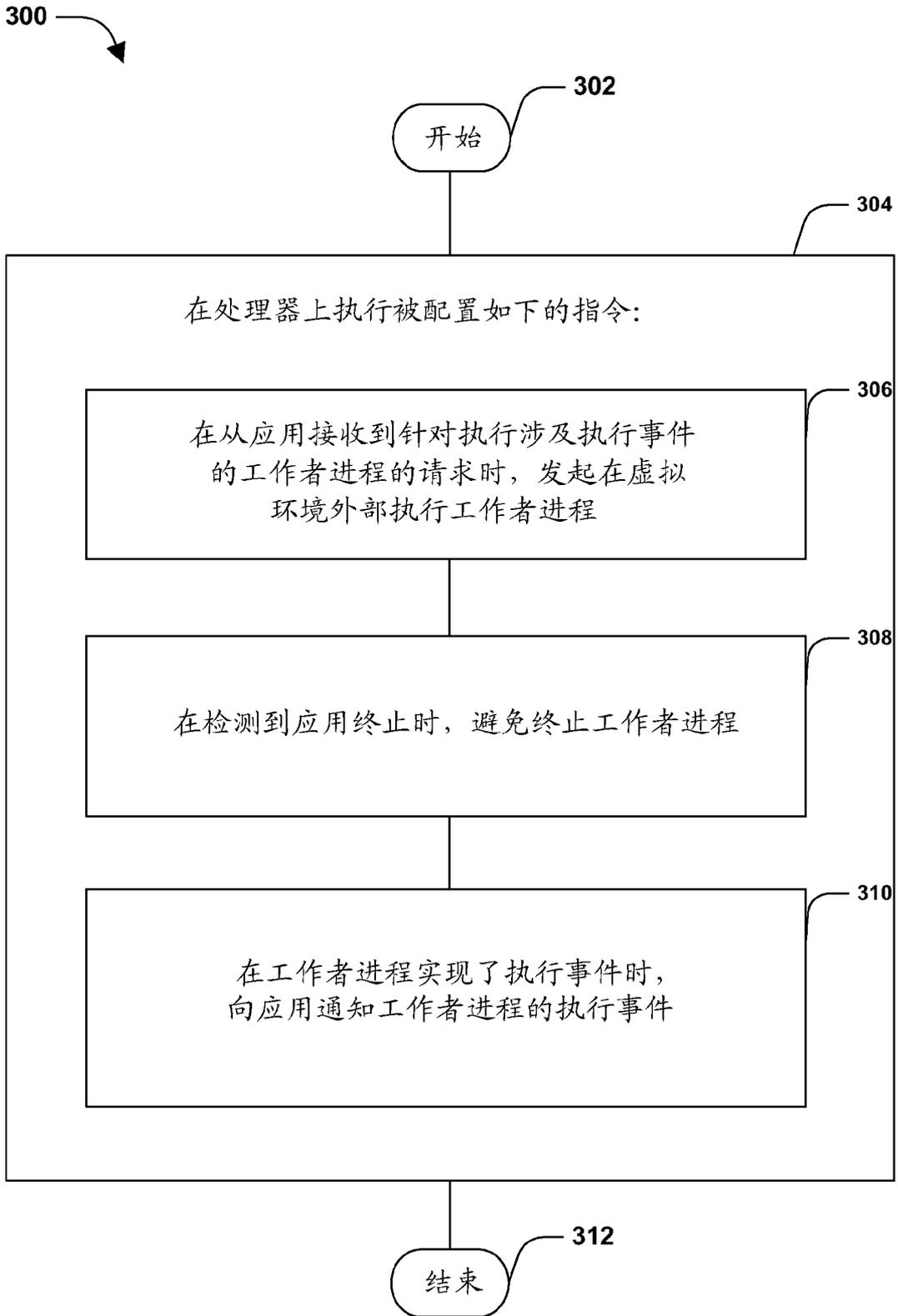


图 3

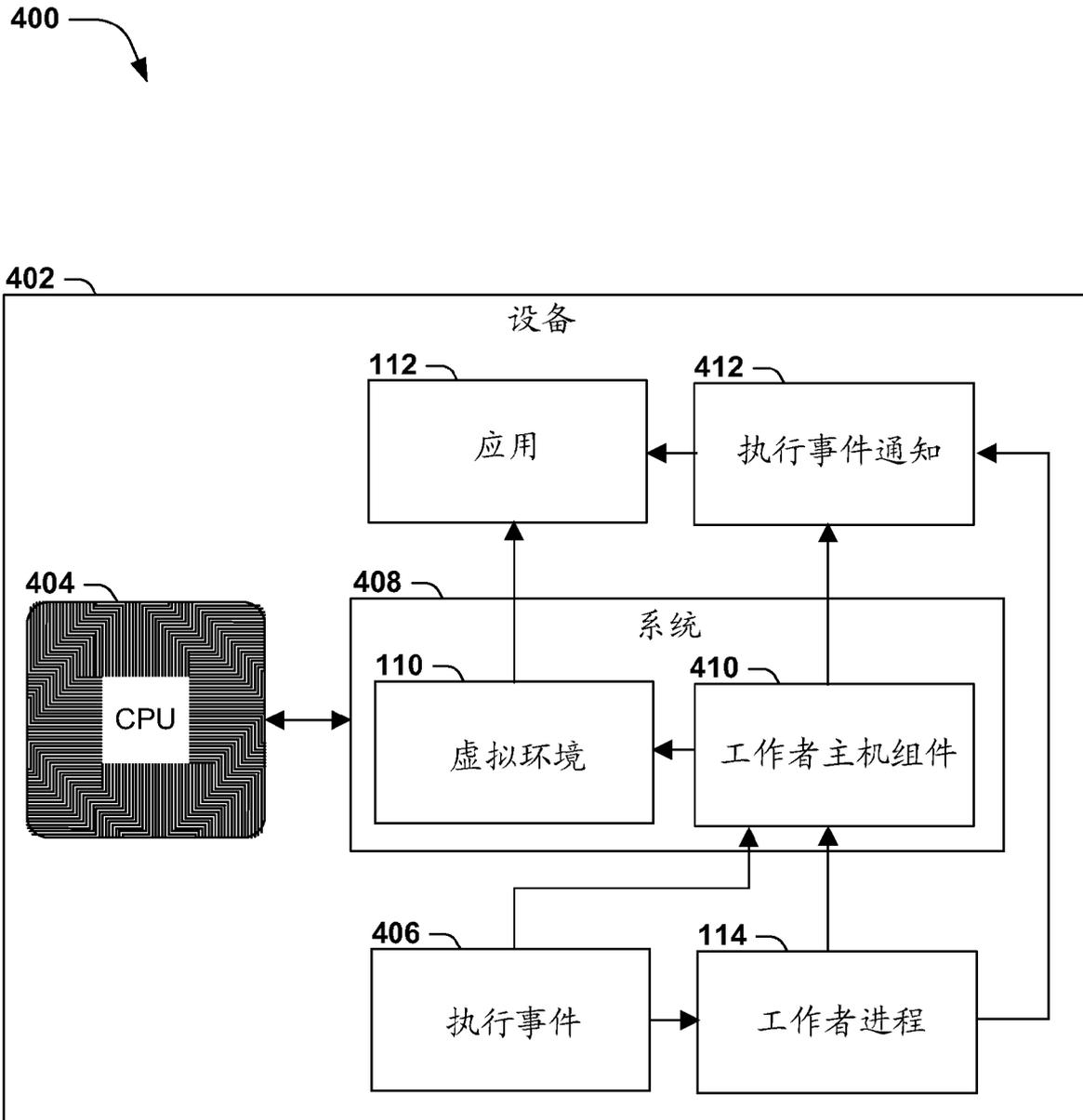


图 4

500

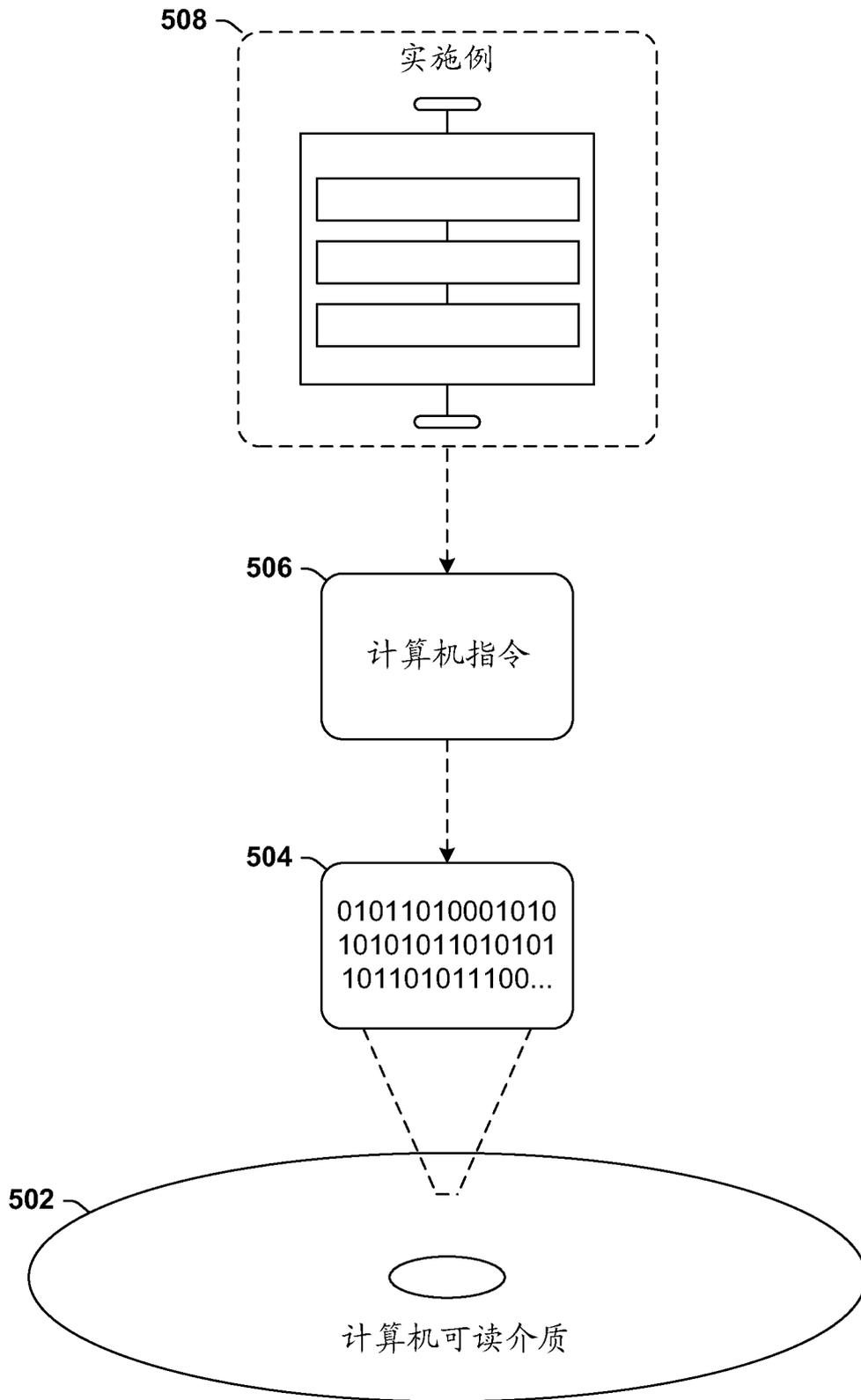


图 5

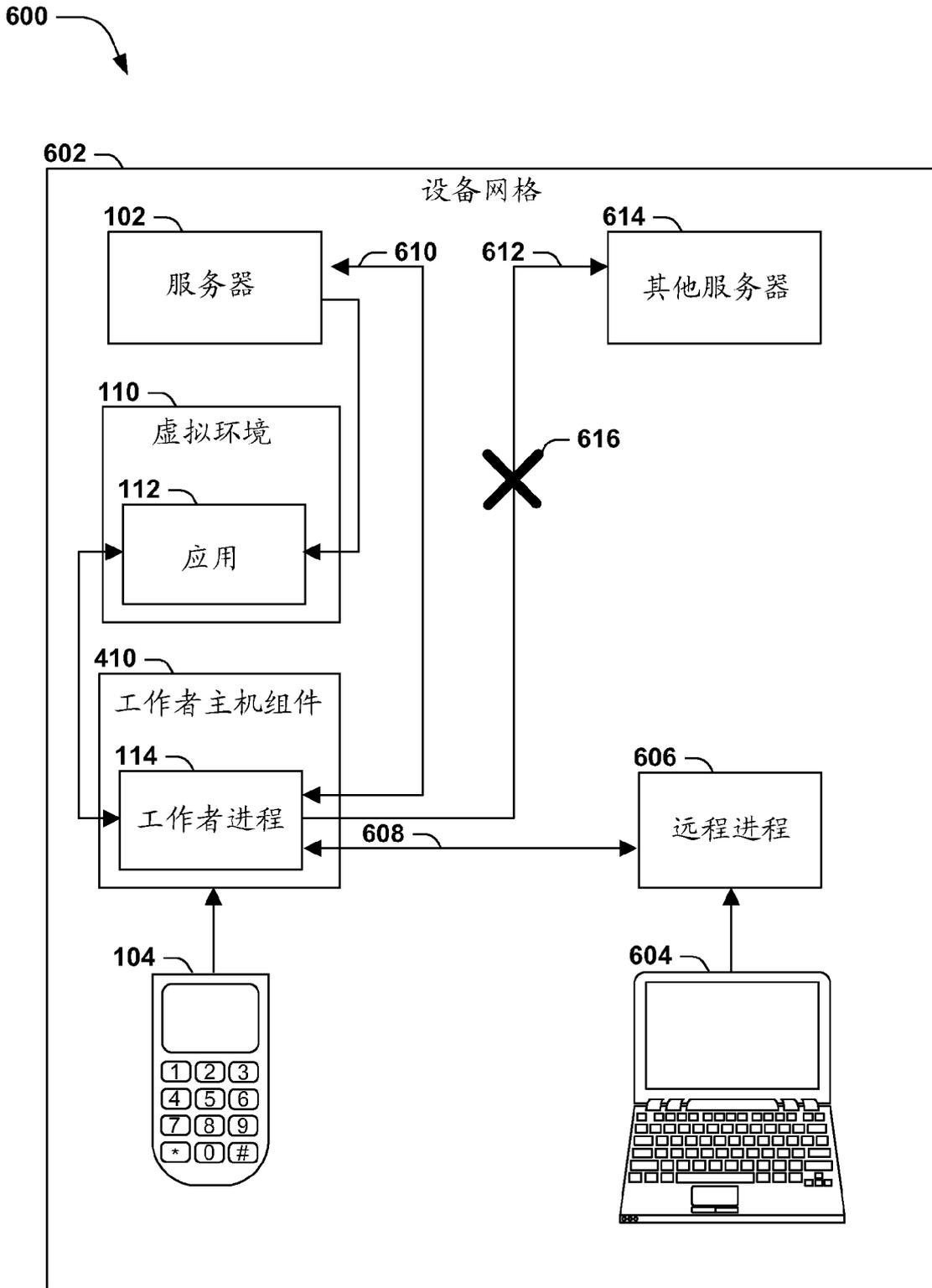


图 6

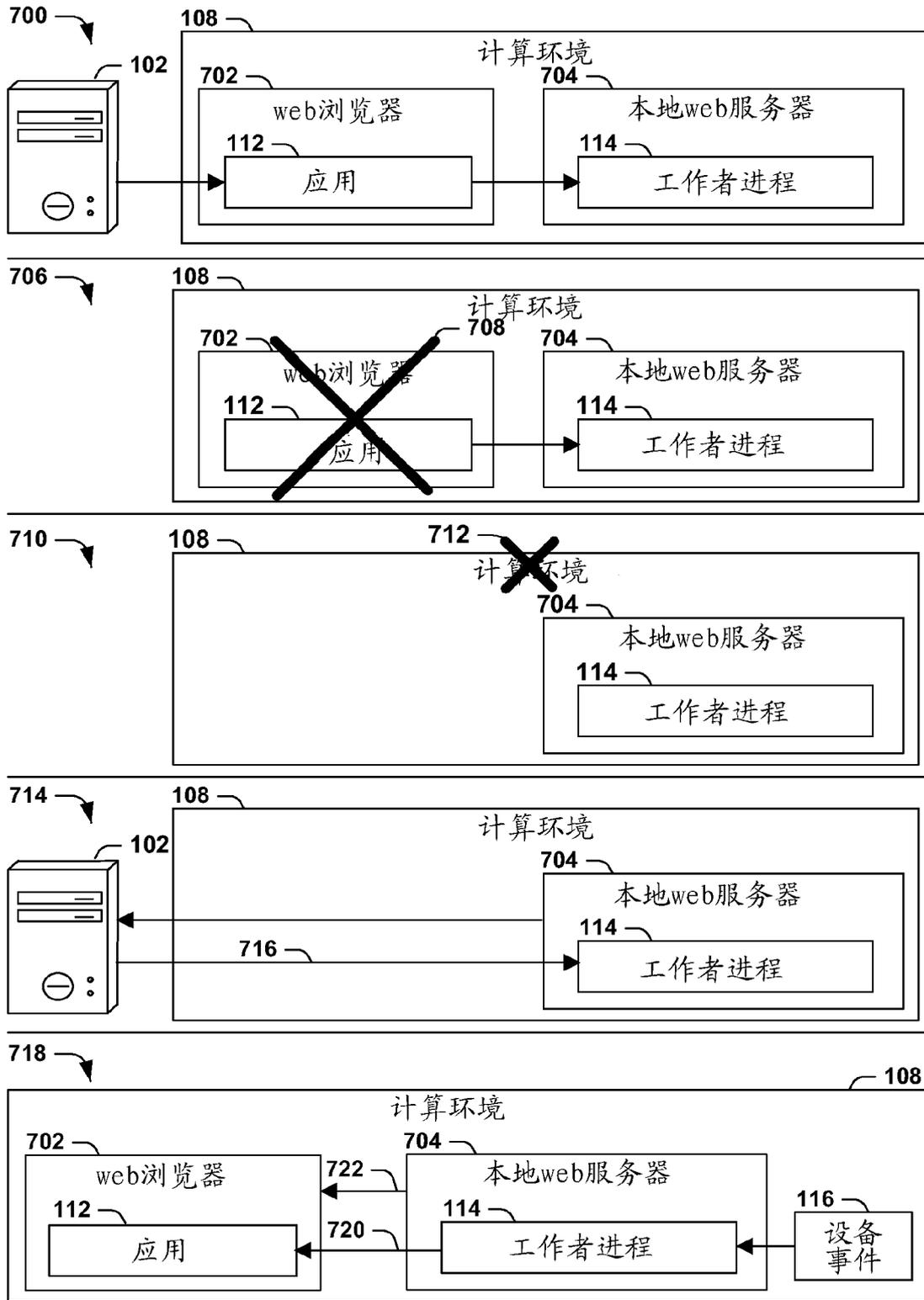


图 7

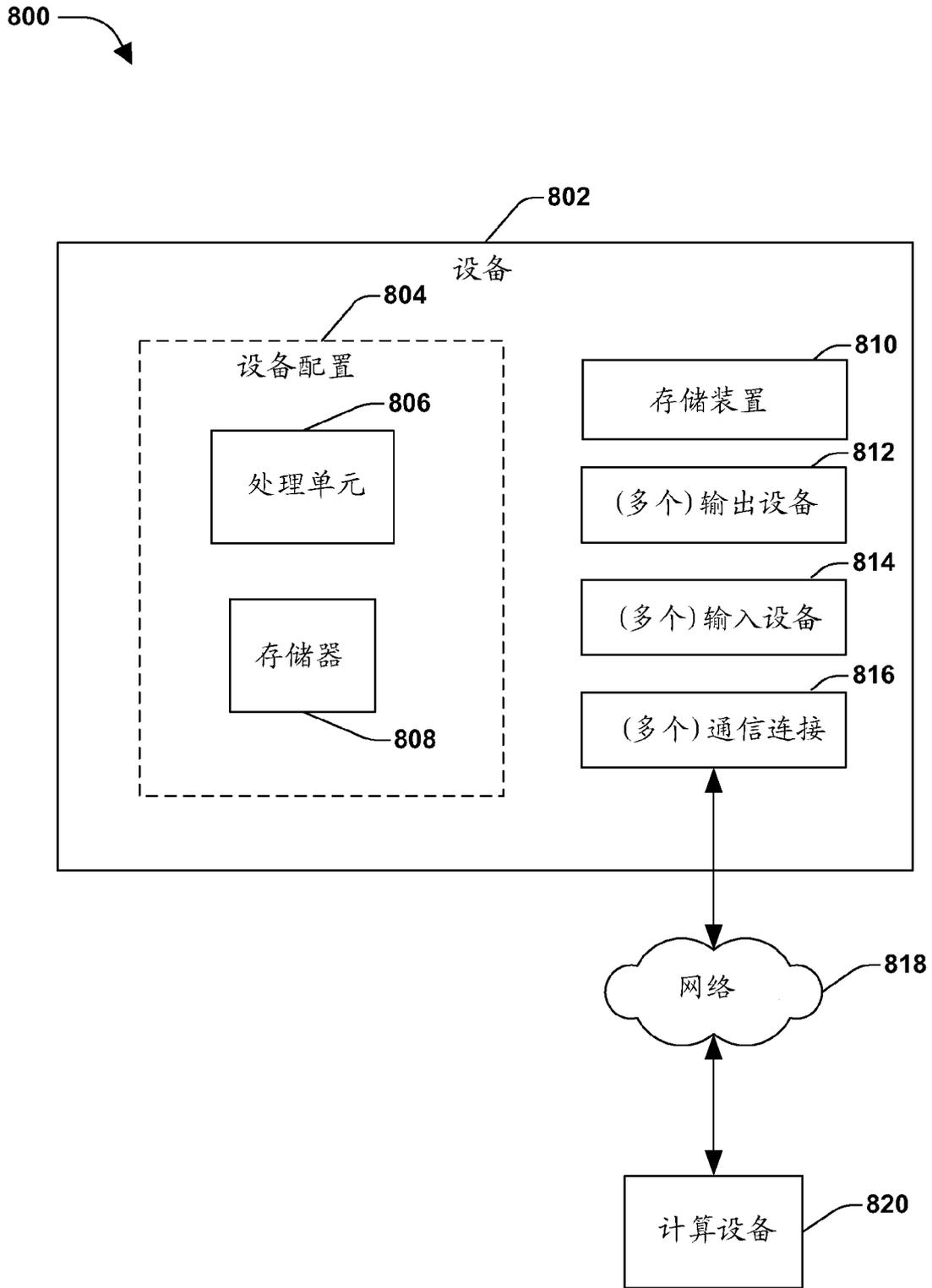


图 8