

(12) 发明专利申请

(10) 申请公布号 CN 103365724 A

(43) 申请公布日 2013. 10. 23

(21) 申请号 201310112286. 8

(22) 申请日 2013. 04. 02

(30) 优先权数据

13/437, 506 2012. 04. 02 US

13/613, 350 2012. 09. 13 US

(71) 申请人 国际商业机器公司

地址 美国纽约

(72) 发明人 M·H·道森 高雨青 伊藤爱

G·约翰逊 S·R·西拉姆

(74) 专利代理机构 北京市中咨律师事务所

11247

代理人 于静 张亚非

(51) Int. Cl.

G06F 9/50 (2006. 01)

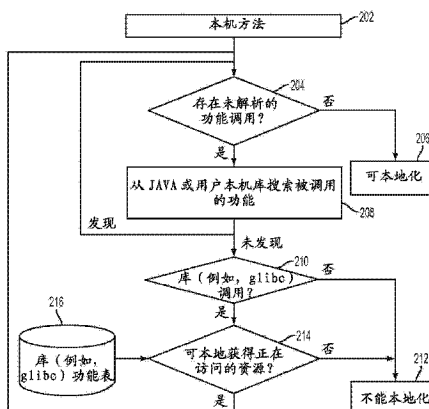
权利要求书2页 说明书8页 附图6页

(54) 发明名称

用于检测可本地化的本机方法的方法和系统

(57) 摘要

本发明涉及一种用于检测可本地化的本机方法的方法和系统。检测可本地化的本机方法可包括静态地分析本机方法的本机二进制文件。对于所述本机二进制文件中调用的每个功能调用，检查通过该功能调用访问的资源是否可本地地获得。如果通过所述本机方法访问的全部资源均可本地地获得，则将该方法标注为可本地化。



1. 一种检测可本地化的本机方法的方法,包括:

由处理器分析本机方法的二进制文件以识别在所述二进制文件中调用的一个或多个功能调用;

对于所识别的一个或多个功能调用中的每一个,由处理器自动判定所述功能调用是否访问无法在选定机器上本地地获得的资源;

如果判定所述一个或多个功能调用均未访问无法在所述选定机器上本地地获得的资源,则将所述本机方法标注为可本地化;以及

如果判定所述一个或多个功能调用中的至少一个功能调用访问无法在所述选定机器上本地地获得的资源,则将所述本机方法标注为不可本地化。

2. 根据权利要求1所述的方法,其中静态地执行所述分析。

3. 根据权利要求1所述的方法,还包括创建与所述本机方法关联的调用图,所述调用图包括分别表示所述一个或多个功能调用的节点,并且判定步骤包括遍历所述调用图。

4. 根据权利要求3所述的方法,其中判定步骤进一步包括判定由所述调用图的一个节点表示的功能调用是否为库功能调用,以及如果该功能调用是库功能调用,则查找库功能表以判定该库功能调用是否访问无法在所述选定机器上本地地获得的资源。

5. 根据权利要求3所述的方法,其中判定步骤进一步包括判定由所述调用图的一个节点表示的功能调用是否为系统调用,以及如果该功能调用是系统调用,则查找系统调用表以判定该系统调用是否访问无法在所述选定机器上本地地获得的资源。

6. 根据权利要求1所述的方法,还包括标注所述二进制文件中的哪个功能调用正在访问无法在所述选定机器上获得的资源。

7. 根据权利要求1所述的方法,还包括基于所述本机方法是否被判定为可本地化来测量所述本机方法与所述选定机器的亲和力。

8. 根据权利要求7所述的方法,还包括记录亲和力测量。

9. 一种用于检测可本地化的本机方法的系统,包括:

处理器;

分析模块,其可操作以分析本机方法的二进制文件以识别在所述二进制文件中调用的一个或多个功能调用,所述分析模块还可操作以对于所识别的一个或多个功能调用中的每一个,自动判定所述功能调用是否访问无法在选定机器上本地地获得的资源,如果判定所述一个或多个功能调用均未访问无法在所述选定机器上本地地获得的资源,则将所述本机方法标注为可本地化,以及如果判定所述一个或多个功能调用中的至少一个功能调用访问无法在所述选定机器上本地地获得的资源,则将所述本机方法标注为不可本地化。

10. 根据权利要求9所述的系统,其中所述分析模块静态地分析所述二进制文件。

11. 根据权利要求9所述的系统,其中所述分析模块进一步创建与所述本机方法关联的调用图,所述调用图包括分别表示所述一个或多个功能调用的节点,并且所述分析模块遍历所述调用图以判定所述功能调用是否访问无法在选定机器上本地地获得的资源。

12. 根据权利要求11所述的系统,其中所述分析模块进一步判定由所述调用图的一个节点表示的功能调用是否为库功能调用,以及如果该功能调用是库功能调用,则查找库功能表以判定该库功能调用是否访问无法在所述选定机器上本地地获得的资源。

13. 根据权利要求9所述的系统,其中所述分析模块进一步基于所述本机方法是否被

判定为可本地化来测量所述本机方法与所述选定机器的亲和力。

14. 根据权利要求 13 所述的系统,其中所述分析模块进一步记录亲和力测量。

## 用于检测可本地化的本机方法的方法和系统

[0001] 相关申请

[0002] 本申请是在 2012 年 4 月 2 日递交的第 13/437,506 号美国专利申请的延续,以引用方式将其并入本文。

### 技术领域

[0003] 本申请一般地涉及计算机、计算机应用以及具有异构平台的计算机系统,更具体地说,涉及在计算机系统中识别可本地化的功能调用。

### 背景技术

[0004] **Java®**虚拟机(JVM)实现支持**Java®**本机接口(JNI)作为允许**Java®**代码调用以 C 和 C++ (本机代码)编写的方法(并且反之亦然)的机制。传统上,随着执行在 Java™ 中编写的代码与本机代码之间转变,这两种代码在同一进程中由同一线程执行。

[0005] 然而,可以构建 JVM,使得本机代码在一个或多个远程执行容器中运行,所述远程执行容器可位于从其执行**Java®**代码的相同或不同的机器上的单独进程中,以致本机代码不知道其正在与 JVM 分离地执行。该分离防止行为不当的本机代码使 JVM 不稳定,并使能在不同于主 JVM 的环境(例如,安全性上下文,比特宽度)中运行本机代码。

[0006] 在分离的 JVM 中,**Java®**与本机代码之间的调用的成本具有更大的开销和等待时间,这导致在可能的情况下需要减小往返行程。对于标准化的 Java 本机接口(JNI),应用通常必须做出多个调用以获取完成动作所需的信息。

[0007] 在传统的 JVM 中,JNI 对于 Java 调用的开销低到足以可接受。然而,在分布式 JVM 的情况下,进行交叉进程 / 交叉机器调用的等待时间比运行所调用方法所需的时间大若干量级。因为 Java 本机接口是标准化的并且所有现有代码均需在没有修改的情况下在分布式 JVM 中运行,所以无法提供更改 API 以允许应用以更有效的方式请求数据的选择。因此,本发明认识到应当以对现有应用透明的方式将往返行程数减小到可能的最少数量。

[0008] 分布式 JVM 概念在混合系统中是相关的。本申请中的混合系统指包含通用和专用计算平台两者的异构分布式系统。一个实例是来自纽约阿蒙克的国际商业机器公司的 zEnterprise 系统,其具有系统 Z、X86 和 Power7 刀片。由于混合系统可用作根据性能优化工作负载的柔性平台,因此越来越多的应用可受益于在混合系统中运行。

[0009] JVM 代理能够加速混合系统上的**Java®**应用,方式为:在单独加速器上运行代理 JVM,这使所有**Java®**方法均在加速器上运行,而本机方法在主机器上运行。为了提高 JVM 代理上的应用性能,本发明认识到本机方法应被本地化以在加速器上运行,这是因为与单系统情况相比,从加速器上运行的**Java®**方法发起的远程本机方法调用需要额外的网络往返行程开销。然而,并非所有本机方法均可被本地化。通过人工阅读源代码来检查本机方法是否能够被本地化需要大量人力成本。如果新近创建或修改了本机方法,则需要再次进行此类检查。

## 发明内容

[0010] 在一个方面,一种用于检测可本地化的本机方法的方法可包括:分析本机方法的二进制文件以识别在所述二进制文件中调用的一个或多个功能调用。该方法还可包括:对于所识别的一个或多个功能调用中的每一个,判定所述功能调用是否访问无法在选定机器上本地地获得的资源。该方法还可包括:如果判定所述一个或多个功能调用均未访问无法在所述选定机器上本地地获得的资源,则将所述本机方法标注为可本地化。该方法还可包括:如果判定所述一个或多个功能调用中的至少一个功能调用访问无法在所述选定机器上本地地获得的资源,则将所述本机方法标注为不可本地化。

[0011] 在一个方面,一种用于检测可本地化的本机方法的系统可包括:分析模块,其可操作以分析本机方法的二进制文件以识别在所述二进制文件中调用的一个或多个功能调用。所述分析模块还可操作以对于所识别的一个或多个功能调用中的每一个,自动判定所述功能调用是否访问无法在选定机器上本地地获得的资源,如果判定所述一个或多个功能调用均未访问无法在所述选定机器上本地地获得的资源,则将所述本机方法标注为可本地化,以及如果判定所述一个或多个功能调用中的至少一个功能调用访问无法在所述选定机器上本地地获得的资源,则将所述本机方法标注为不可本地化。

[0012] 还提供了存储可由机器执行以执行在此描述的一个或多个方法的指令程序的计算机可读存储介质。

[0013] 下面参照附图详细描述进一步的特征以及各实施方式的结构和操作。在附图中,相同的附图标记表示相同或功能相似的元素。

## 附图说明

[0014] 图 1 示出本发明的一个实施方式中的混合系统内的主处理器和单独加速器的一个实例;

[0015] 图 2 是示出本发明一个实施方式中的判定功能是否可本地化的方法的流程图;

[0016] 图 3 示出了实例 x86 本机方法;

[0017] 图 4 示出了本发明一个实施方式中的针对本机方法创建的调用图的一个实例;

[0018] 图 5A 和图 5B 示出了本发明一个实施方式中的虚拟机模式(schema);

[0019] 图 6 示出了本发明一个实施方式中的可实现可本地化功能识别系统的实例计算机或处理系统的示意图。

## 具体实施方式

[0020] 本发明公开了用于识别可本地化功能或方法调用的方法。图 1 示出了混合系统 100 的主服务器 102 和单独加速器(处理核心) 104 的一个实例。通过在单独加速器 104 上运行代理 JVM108,诸如 JVM 代理的代理 106 可在混合系统 100 上加速应用(例如, **Java®** 应用)。例如,主服务器 102 上的代理 JVM106 可拦截应用调用,并将调用发送到加速器 104,以便代理 JVM108 运行应用调用 110。在本发明的一个实施方式中,例如,方法自动识别可在加速器 104 处本地化哪些应用调用,使得主服务器 102 上的代理 JVM106 可将这些应用调用发送到代理 JVM108。在本发明的一个实施方式中,不使用远程资源(例如, 114)或仅使用本

地资源(116)的那些应用调用(本地的本机方法 112)可被识别为可在加速器上本地化。可使访问加速器之外的资源的本机方法 118 在主服务器 102 或另一处理器上运行。尽管图 1 示出了 JVM 和 **Java®** 方法,但应理解,本发明的实施方式可应用于任何其它混合系统。

[0021] 为了确保应在加速器上获得本机方法访问的所有本机资源,在一个实施方式中,本发明的方法自动地和静态地检查本机方法的本地性。

[0022] 本机方法指这样的计算机代码:一旦代码被编译,代码就是机器相关的,例如,依赖于为其编译代码的机器的固定特性。本机方法通常在平台特定的编译器中进行编译,例如,C 或 C++,或其它计算机编程语言编译器。本机资源指硬件和 / 或附属于特定机器的处理元件,例如,网卡、存储资源、机器或处理器的加密 / 解密功能、本地存储的数据等。

[0023] 在本发明的一个实施方式中,可以检测本机方法中诸如 glibc 调用的库调用,并可确定它们访问的资源的本地可获得性。例如,通过分析编译后的代码(例如,本地“.so”二进制文件)并创建用于本机方法的调用图,可静态地确定本机方法对资源的访问。如果在本机方法及其所有被调用功能中没有标准库调用,则将本机方法标记为可本地化。在之前步骤中没有标记为可本地化的本机方法中,如果在该方法或其被调用功能中存在调用一个或多个标准库调用的本机方法,并且由它们访问的全部本机资源可在本地平台(例如,加速器)上本地地获得,则将本机方法标记为可本地化。对于在之前步骤中没有标记为可本地化的本机方法,将它们标记为不可本地化。可以例如在不运行编译后的代码的情况下离线地执行所述代码的静态分析。

[0024] 图 2 是示出判定功能是否可本地化以确保可本地获得本机方法使用的所有本机资源的方法的流程图。通常,方法或功能经由标准的库功能访问本机资源。参照图 2,在步骤 202,分析诸如二进制文件的编译后的代码,并可创建每个本机方法的调用图。调用图展示了本机方法调用的功能或方法序列,以及这些功能和方法又可调用的功能或方法,等等;调用图中的节点表示功能调用。通过遍历为本机方法创建的调用图,在步骤 204,判定是否存在未解析的功能调用,例如,库调用或另一功能调用。如果没有未解析的功能调用,在步骤 206,确定本机方法为可本地化的。如果存在功能调用以在本机方法中进一步分析(例如,经由调用图确定),则在步骤 208 从程序编译器库或用户本机库搜索被调用的功能。例如,在一个实施方式中,本发明的逻辑可首先搜索 JVM 和用户库。如果本机方法的功能调用序列位于库中,它意味着本机方法不需要外部资源,从而在一个实施方式中,本发明的逻辑可确定本机方法是可本地化的。如果在程序编译器或用户本机库中发现该功能,则逻辑继续到步骤 204。否则,在步骤 210,判定功能调用是否为标准库调用。如果否,在步骤 212,判定本机方法不可本地化,例如,由于没有在本地发现功能调用。如果在步骤 210,判定功能调用是标准库调用,逻辑继续到步骤 214。在步骤 214,判定是标准库调用的功能是否正在访问可本地获得的资源。这可通过查找具有库调用的数据库表 216 等进行。表 216 例如可包括有关库调用正在访问哪些资源的信息。如果功能正在访问可本地获得的资源,逻辑继续到步骤 204,其中检查另一调用图节点。否则,在步骤 212 将本机方法确定为不可本地化。

[0025] 本发明的一个实施方式提供了库功能表 216 (下面展示用于 glibc 标准库的一个实例),其包括关于本机资源和本地可获得性的标准库功能信息。此类库功能表 216 可用于检查由标准库功能访问的本机资源及其本地可获得性。下面的实例示出了此类用于 glibc 标准库的表。然而,本发明并不将该表仅限于 glibc 标准库。如果存在从本机方法或其没

有调用标准库功能的被调用功能所直接调用的系统调用,则本发明的一个实施方式可使用系统调用表,系统调用表包括其访问资源和本地可获得性,与标准的库功能表相似。

[0026] 表 1:glibc 功能表。

[0027]

〈功能名称〉	〈本机资源〉	〈本地可获得性〉
Func1	套接字	本地可获得
Func2	文件	不可获得
Func3	无	
...		

[0028] 图 3 示出了来自在 **Intel®** 8086 中央处理单元(CPU)的指令集架构中编译的典型 x86 二进制的两个示例性本机方法。在该实例中,编译后的二进制可包括两个本机方法“methodA”302 和“methodE”306。此外,其包括“functionB”304 和“functionC”308,“functionD”310。图 4 示出了用于该实例的示例调用图。用于该实例的调用图可包括表示“methodA”402、“functionB”414 和“functionE”412 的节点。“methodA”402 具有子节点“functionB”404,“functionC”406 和“functionD”408。此外,“functionD”408 调用图可包括“callE”410 作为子节点。本机“methodE”412 不包括子节点,从而其调用图没有子代。

[0029] 在该实例中,本发明的方法评估本机 MethodE306,并确定其可本地化,因为在该方法中没有其它功能调用。对于“methodA”302,该方法调用三个功能,“functionB”、“functionC”和“functionD”。在该实例中,“functionB”304 是内部功能,并且其没有子代。“functionC”308 是外部功能,在该实例中引用外部库。这意味着“functionC”是 libB. so 中的库外部功能。在该实例中,libA. so 和 libB. so 都是 JVM 或用户库。如果“methodA”没有调用“functionD”,则“methodA”可被本地化。在该实例中“functionD”310 是标准的库功能。在一个实施方式中,本发明的方法评估“functionB”304,并确定它是没有任何其他功能调用的内部功能。在一个实施方式中,本发明的方法评估“functionC”308,并确定它是外部功能,从而将分析其相应的库。在这种情况下,检查该功能是否具有功能调用,与库内部功能的方式相同。如果与“functionC”名称相同的功能存在于多个库中,在一个实施方式中,本发明的方法可发现库以根据系统链接参数或其它内容进行检查。如果不能确定要检查的库,则可检查与“functionC”名称相同的全部功能,并判定它们是否可本地化,以便判定“functionC”是否能够被本地化。

[0030] 在一个实施方式中,本发明的方法评估“functionD”310 并确定它是对调用系统调用“callE”的标准库的调用。然后,进行检查以判定该系统调用(在该实例中,“callE”)是否访问本机资源或本地机器(在该实例中,x86CPU)外部的资源。如果是,则确定“methodA”302 可本地化。

[0031] 在本发明的一个实施方式中,可进一步将没有确定为“可本地化”的那些方法分类为“可以本地化”或“不可本地化”。如果将本机方法确定或分类为可本地化,则该方法被视

为与编译本机方法以在其上运行的本地机具有亲和力(affinity)。否则,该方法被视为与远程机器具有亲和力。可相应地创建亲和力表或数据库,并可基于外部环境(例如将资源添加到本地机器和 / 或远程机器,修改本机方法等)而更新或更改该表。

[0032] 图 5A 和图 5B 示出了虚拟机进程和线程模式。在传统的 **Java**<sup>®</sup>虚拟机中,单个 **Java**<sup>®</sup>进程可执行 **Java**<sup>®</sup>代码和本机代码,如 502 所示。在分布式 **Java**<sup>®</sup>虚拟机中,本机代码可在与运行 **Java**<sup>®</sup>代码的进程 506 分离的远程执行容器 504 中运行。在一个实施方式中,本发明的方法可识别可在远程执行容器中本地化的本机方法 508,从而减少交叉机器 / 交叉进程调用。

[0033] 图 6 示出了本发明一个实施方式中的可实现可本地化功能识别系统的示例性计算机或处理系统的示意图。计算机系统仅是合适的处理系统的一个实例,并不意在对这里所描述方法的实施方式的使用或功能范围进行任何限定。所示出的处理系统可与多个其它通用或专用计算系统环境或配置一起工作。适于与图 6 中所示的处理系统一起使用的公知计算系统、环境和 / 或配置的实例可包括(但不限于)个人计算机系统、服务器计算机系统、瘦客户机、厚客户机、手持或膝上型设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费电子设备、网络 PC、微计算机系统、大型机计算机系统和包括任何上述系统或设备的分布式云计算环境,等等。

[0034] 可在由计算机系统执行的计算机系统可执行指令(例如,程序模块)的通用上下文中描述计算机系统。通常,程序模块可包括例程、程序、对象、组件、逻辑、数据结构、以及执行特定任务或实现特定抽象数据类型的其它内容。计算机系统可在分布式云计算环境中实现,其中任务由通过通信网络链接的远程处理设备执行。在分布式云计算环境中,程序模块可位于本地或包括记忆存储设备的远程计算机系统存储介质中。

[0035] 计算机系统的组件可包括,但不限于,一个或多个处理器或处理单元 12、系统存储器 16、以及将包括系统存储器 16 的各系统组件与处理器 12 耦合的总线 14。处理器 12 可包括执行在此所描述方法的本地化功能识别模块 10。模块 10 可编程到处理器 12 的集成电路中,或从存储器 16、存储设备 18、网络 24 或其组合加载。

[0036] 总线 14 可表示若干类型的总线结构中的任意一个或多个,包括存储总线或存储控制器、外围总线、加速图形端口、和使用多个总线架构中任一个的处理器或本地总线。作为实例而非限制,此类架构包括工业标准架构 (ISA) 总线、微通道架构 (MAC) 总线、增强 ISA (EISA) 总线、视频电子标准协会 (VESA) 本地总线、和外围元件互连 (PCI) 总线。

[0037] 计算机系统可包括各种计算机系统可读介质。此类介质可以是可由计算机系统存取的任何可用介质,并且其可包括易失性和非易失性介质、可移除和非可移除介质。

[0038] 系统存储器 16 可包括易失性存储器形式的计算机系统可读介质,例如随机存取存储器 (RAM) 和 / 或高速缓冲存储器等。计算机系统可还包括其他可移除 / 非可移除、易失性 / 非易失性计算机系统存储介质。仅作为实例,可提供存储系统 18 以便从非可移除、非易失性磁介质(例如,“硬盘驱动器”)读取或写入非可移除、非易失性磁介质(例如,“硬盘驱动器”)。尽管未示出,但是可提供用于从可移除、非易失性磁盘(例如,“软盘”)读取或写入可移除、非易失性磁盘(例如,“软盘”)的磁盘驱动器,以及用于从可移除、非易失性光盘(诸如 CD-ROM、DVD-ROM 或其他光介质)读取或写入可移除、非易失性光盘(诸如 CD-ROM、DVD-ROM 或其他光介质)的光盘驱动器。在处理情况下,它们均可通过一个或多个数据介质接口与总



线 14 连接。

[0039] 计算机系统还可与诸如键盘、指点设备、显示器 28 等的一个或多个外部设备 26、与能够使用户与计算机系统交换的一个或多个设备、和 / 或能够使计算机系统与一个或多个其它计算机设备进行通信的任何设备(例如,网卡、调制解调器等)进行通信。这样的通信可通过输入 / 输出(I/O)接口 20 发生。

[0040] 同时,计算机系统可经由网络适配器 22 与一个或多个网络 24(例如局域网(LAN)、通用广域网(WAN)、和 / 或公共网络(例如,互联网))进行通信。如所描述的,网络适配器 22 通过总线 14 与计算机系统的其它组件进行通信。可以理解的是,尽管没有示出,其它硬件和 / 或软件组件可结合计算机系统使用。实例包括但不限于:微代码、设备驱动器、冗余的处理单元、外部磁盘驱动阵列、RAID 系统、磁带驱动器、和数据档案存储系统等。

[0041] 所属技术领域的技术人员知道,本发明可以实现为系统、方法或计算机程序产品。因此,本发明的各方面可以具体实现为以下形式,即:可以是完全的硬件、也可以是完全的软件(包括固件、驻留软件、微代码等),还可以是硬件和软件结合的形式,本文一般称为“电路”、“模块”或“系统”。此外,在一些实施例中,本发明的各方面还可以实现为在一个或多个计算机可读介质中的计算机程序产品的形式,该计算机可读介质中包含计算机可读的程序代码。

[0042] 可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是一但不限于一电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM 或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0043] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括一但不限于一电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0044] 计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括一但不限于一无线、电线、光缆、RF 等等,或者上述的任意合适的组合。

[0045] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明的各方面的操作的计算机程序代码,所述程序设计语言包括面向对象的程序设计语言—诸如 Java、Smalltalk、C++,还包括常规的过程式程序设计语言—诸如“C”语言或类似的设计语言,诸如 Perl、VBS 或相似语言的脚本语言,和 / 或诸如 Lisp 和 ML 的功能语言以及诸如 Prolog 的面向逻辑的语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络—包括局域网(LAN)或广域网(WAN)—连接到用户计算机,或者,可以连接

到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0046] 将参照本发明实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述本发明的各方面。应当理解,流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合,都可以由计算机程序指令实现。这些计算机程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器,从而生产出一种机器,这些计算机程序指令通过计算机或其它可编程数据处理装置执行,产生了实现流程图和/或框图中的方框中规定的功能/操作的装置。

[0047] 也可以把这些计算机程序指令存储在能使得计算机或其它可编程数据处理装置以特定方式工作的计算机可读介质中,这样,存储在计算机可读介质中的指令就产生出一个包括实现流程图和/或框图中的方框中规定的功能/操作的指令装置(instruction means)的制品(manufacture)。

[0048] 也可以把计算机程序指令加载到计算机、其它可编程数据处理装置、或其它设备上,使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤,以产生计算机实现的过程,从而使得在计算机或其它可编程装置上执行的指令能够提供实现流程图和/或框图中的方框中规定的功能/操作的过程。

[0049] 附图中的流程图和框图显示了根据本发明的多个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分,所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0050] 计算机程序产品可包括能够实现这里所描述方法的全部相应特征,并且在装入计算机系统时,能够执行该方法。在当前上下文中,计算机程序、软件程序、程序、或软件指一组指令的以任何语言、代码或符号表示的任何表达,旨在使具有信息处理能力的系统直接执行特定的功能,或者执行以下两者之一或全部后执行特定的功能:a)转换为另一种语言、代码或符号;和/或 b)以不同的材料形式再现。

[0051] 这里使用的术语仅用于描述特定的实施方式,并不意在对本发明进行限制。如在这里所使用的,单数形式“一”、“一个”和“该”意在包括复数形式,除非在上下文中明确地进行指示。还可以进一步理解的是,当术语“包含”、“由...组成”在此说明书中使用,说明存在所声明的特征、整数、步骤、操作、元素和/或元件,但是不排除存在或增加一个或多个其它特征、整数、步骤、操作、元素、元件和/或其组合。

[0052] 以下的权利要求中的对应结构、材料、操作以及所有功能性限定的装置(means)或步骤的等同替换,旨在包括任何用于与在权利要求中具体指出的其它单元相组合地执行该功能的结构、材料或操作。所给出的对本发明的描述其目的在于示意和描述,并非是穷尽性的,也并非是要将本发明限定到所表述的形式。对于所属技术领域的普通技术人员来说,在不偏离本发明范围和精神的情况下,显然可以作出许多修改和变型。对实施例的选择和

说明,是为了最好地解释本发明的原理和实际应用,使所属技术领域的普通技术人员能够明了,本发明可以有适合所要的特定用途的具有各种改变的各种实施方式。

[0053] 本发明的各方面可体现为在计算机或机器可用或可读介质中包含的程序、软件或计算机指令,在这种情况下,当在计算机、处理器和 / 或机器上运行时,计算机或机器执行该方法的步骤。在本发明中还提供了通过机器可读的程序存储设备,其有形地体现可由机器执行的指令程序以执行所描述的各种功能和方法。

[0054] 本发明的系统和方法可在通用计算机或专用计算机系统上执行或运行。在本申请中使用的术语“计算机系统”和“计算机网络”可包括固定的和 / 或便携式计算机硬件、软件、外围设备和存储设备的各种组合。计算机系统可包括连成网络或以其他方式链接以协作执行的多个单个组件,或可包括一个或多个独立的组件。本申请的计算机系统的硬件和软件组件可包括和可被包括在诸如桌上型计算机、膝上型计算机和 / 或服务器之类的固定和便携式设备中。模块可以是设备、软件、程序、或执行一些“功能”的系统的组件,其可体现为软件、硬件、固件、电子电路等。

[0055] 上面描述的实施方式是解释性的实例,不应将其解释为将本发明限于这些特定的实施方式。因此,在不偏离在所附权利要求中限定的本发明的精神或范围的情况下,本领域技术人员可实现各种变化和修改。

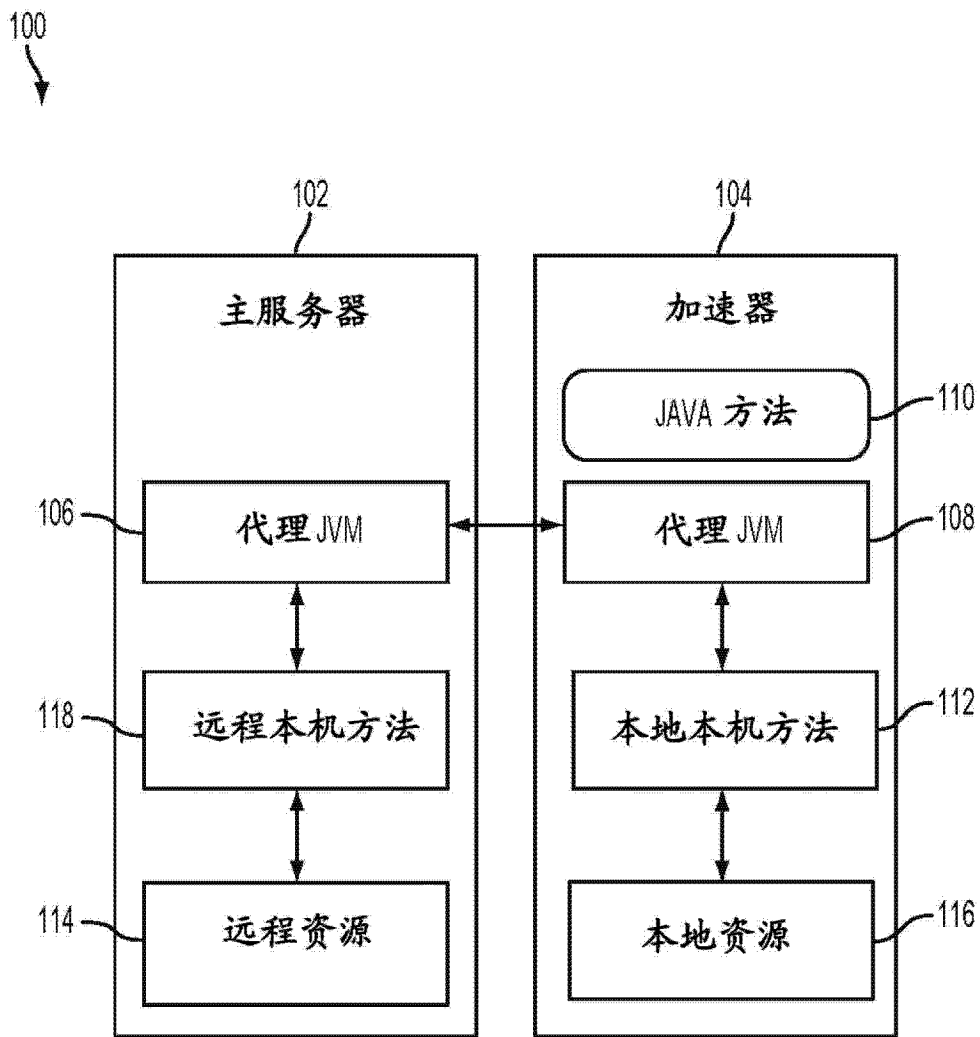


图 1

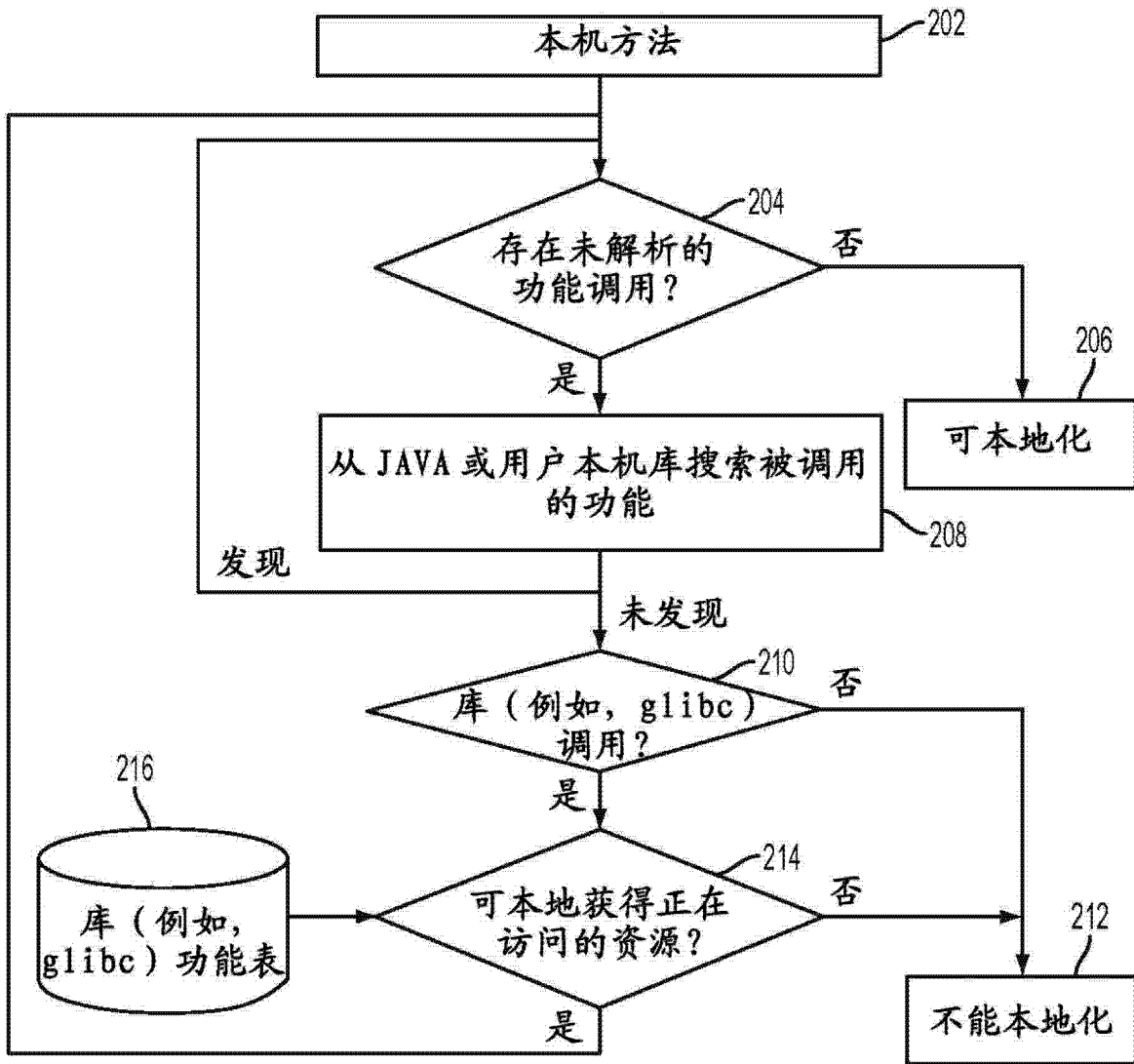


图 2

示例性本机方法

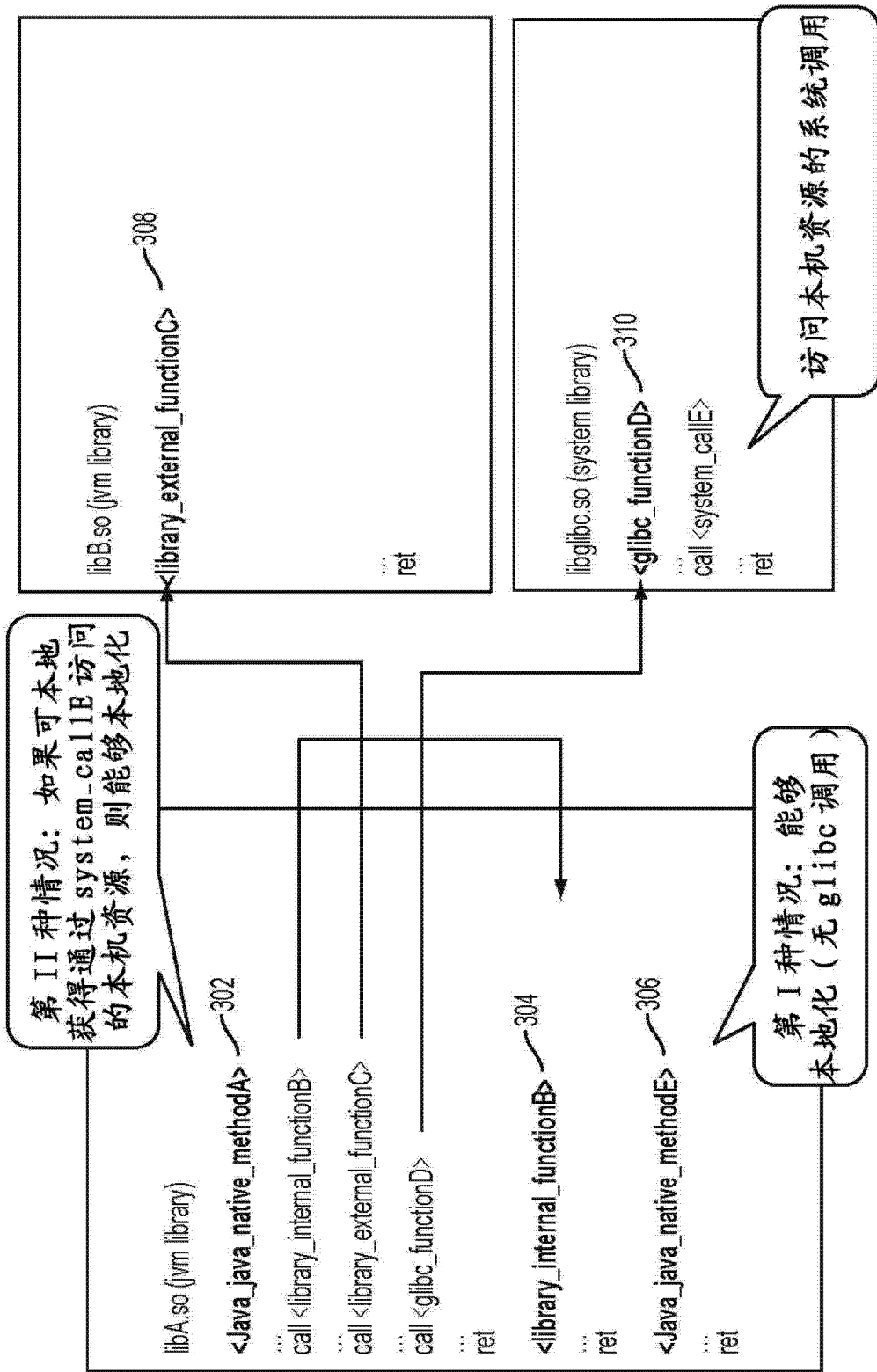


图 3

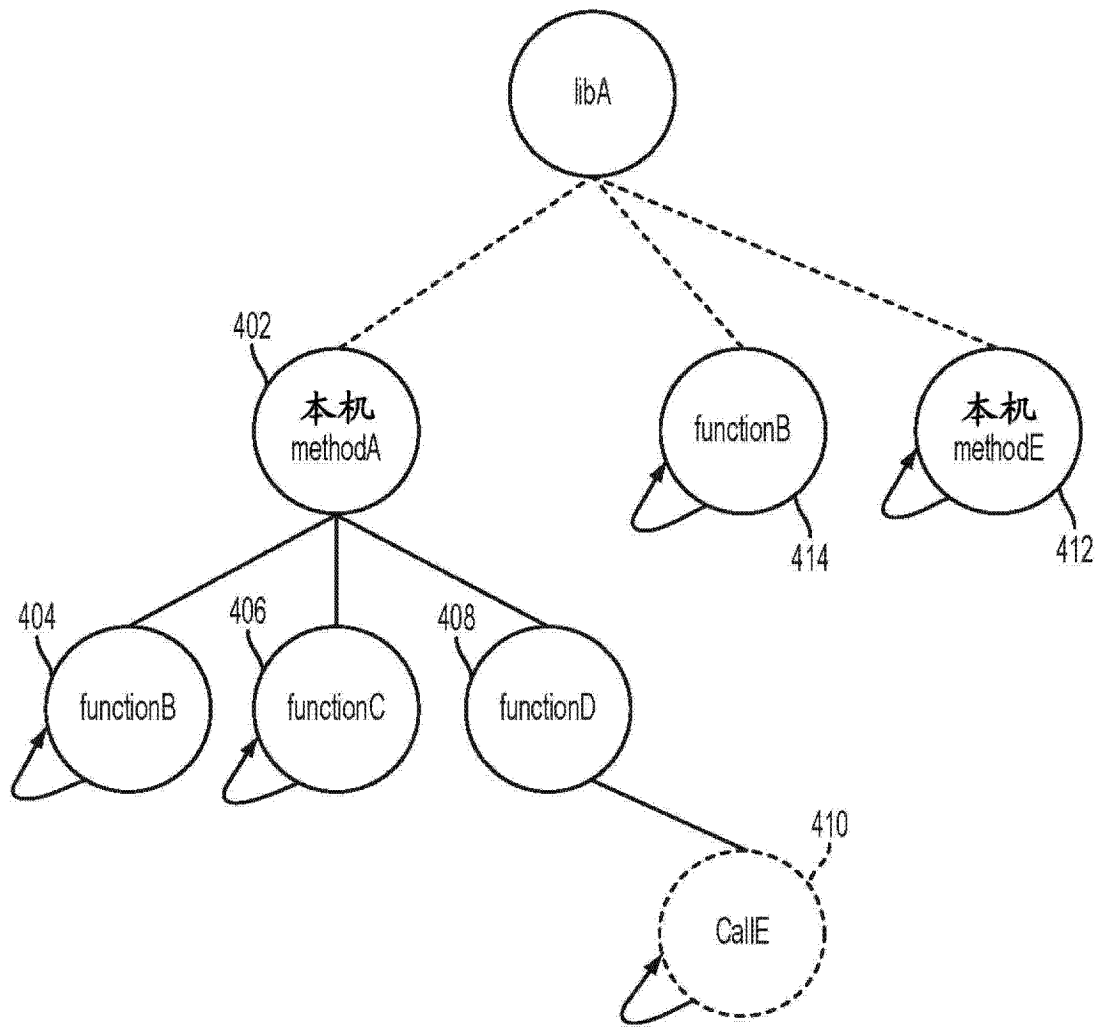


图 4

传统 JAVA 虚拟机

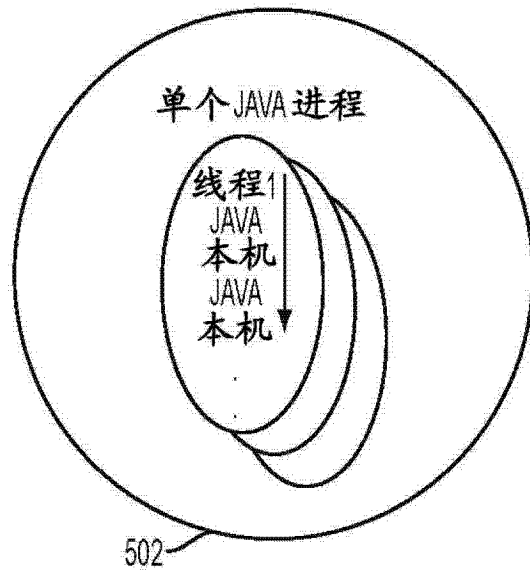


图 5A

分布式 JAVA 虚拟机

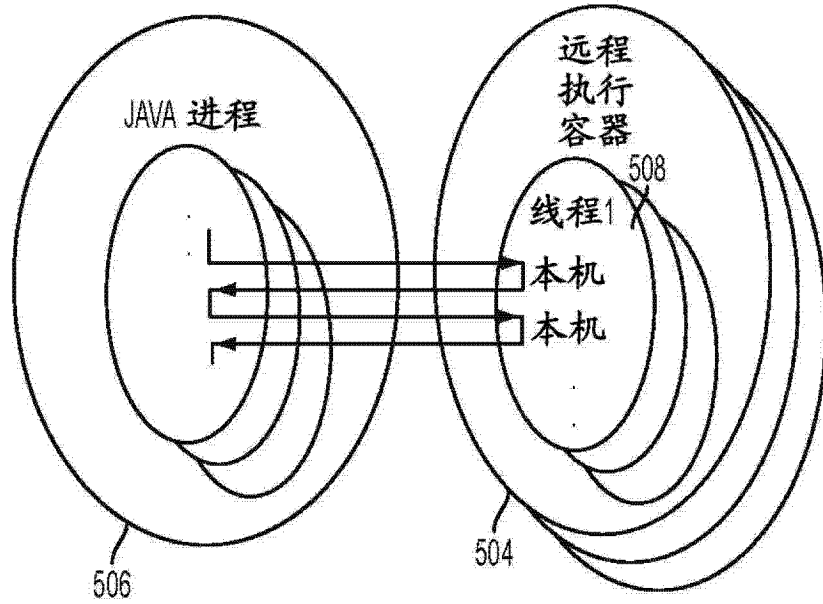


图 5B



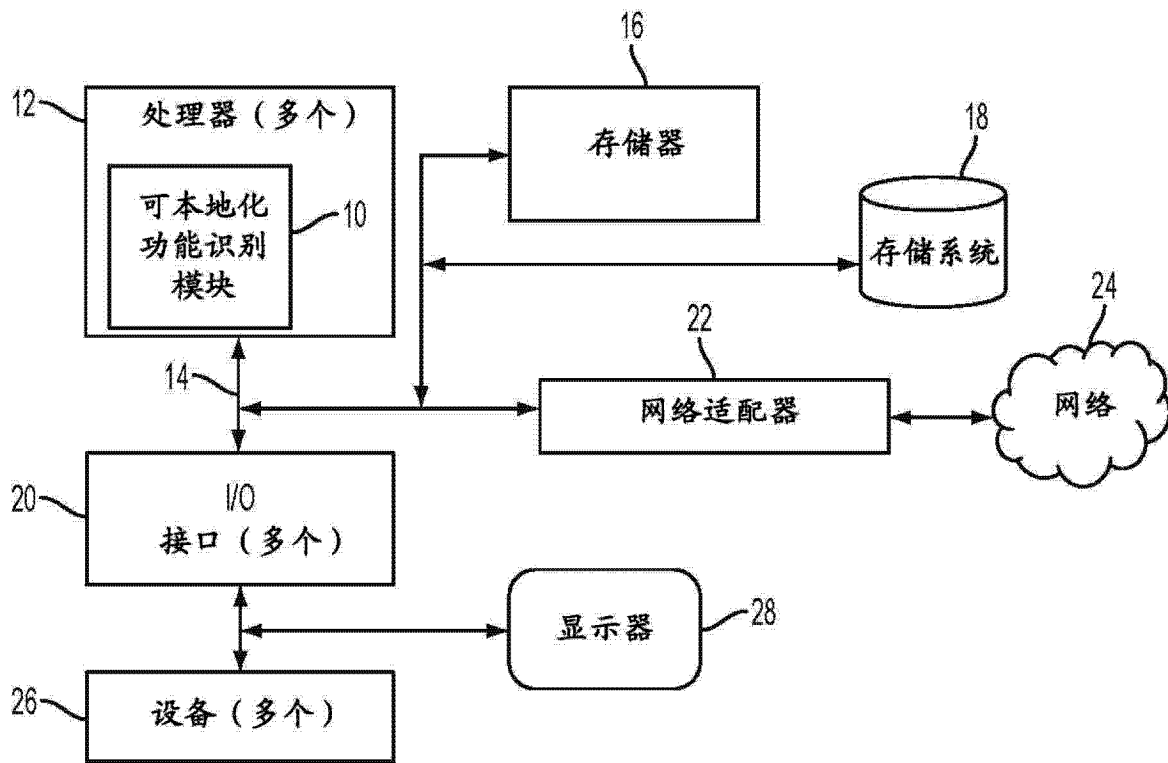


图 6