



(12) 发明专利

(10) 授权公告号 CN 107408053 B

(45) 授权公告日 2021.04.16

(21) 申请号 201580078466.6

(22) 申请日 2015.04.10

(65) 同一申请的已公布的文献号
申请公布号 CN 107408053 A

(43) 申请公布日 2017.11.28

(85) PCT国际申请进入国家阶段日
2017.09.28

(86) PCT国际申请的申请数据
PCT/IB2015/000883 2015.04.10

(87) PCT国际申请的公布数据
W02016/162720 EN 2016.10.13

(73) 专利权人 谷歌有限责任公司
地址 美国加利福尼亚州

(72) 发明人 叶夫根尼·艾尔钦

尼古拉·伊戈蒂

安德烈·哈尔佳温

德米特里·波卢欣

(74) 专利代理机构 中原信达知识产权代理有限
责任公司 11219

代理人 李宝泉 周亚荣

(51) Int. Cl.

G06F 8/52 (2018.01)

G06F 9/455 (2006.01)

(56) 对比文件

CN 104462962 A, 2015.03.25

US 2002029308 A1, 2002.03.07

CN 101145109 A, 2008.03.19

审查员 屈姗姗

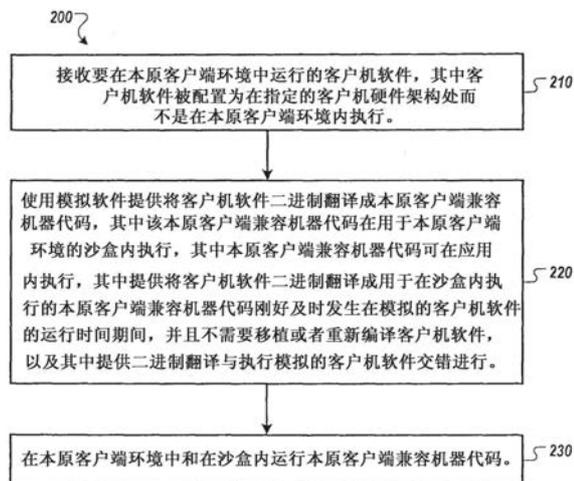
权利要求书3页 说明书11页 附图4页

(54) 发明名称

用于到本原客户端的二进制翻译的方法、系统和介质

(57) 摘要

公开了一种用于二进制翻译的系统和方法。在一些实现方式中,接收要在本原客户端环境中运行的客户机软件。该客户机软件被配置为执行在指定客户机硬件架构处而不是执行在该本原客户端环境内。使用模拟软件提供该客户机软件到本原客户端兼容机器代码的二进制翻译。该本原客户端兼容机器代码在用于该本原客户端环境的沙盒内执行。该本原客户端兼容机器代码能够在应用内执行。提供该客户机软件到用于在该沙盒内执行的该本原客户端兼容机器代码的二进制翻译刚好及时发生在该模拟的客户机软件的运行时间期间,并且不需要移植或者重新编译该客户机软件。提供该二进制翻译与执行该模拟的客户机软件交错进行。



1. 一种用于二进制翻译的方法,包括:

提供用于在主机硬件架构中运行本原客户端软件的本原客户端环境,其中在所述主机硬件架构上运行的主机软件对所述本原客户端软件不可访问;

接收要在本原客户端环境中运行的客户机软件,其中所述客户机软件被配置为执行在指定客户机硬件架构处而不是执行在所述主机硬件架构处或所述本原客户端环境内,以及其中所述指定客户机硬件架构不同于所述主机硬件架构和所述本原客户端环境;

在所述本原客户端环境内与所述本原客户端环境被配置为在其上运行的所述主机硬件架构隔离地生成在所述主机硬件架构中使用的硬件的第一组软件表示和在所述客户机硬件架构中使用的硬件的第二组软件表示;以及

使用模拟软件提供所述客户机软件到用于与所述本原客户端软件一起在所述本原客户端环境内执行并且不能访问在所述主机硬件架构上运行的所述主机软件的本原客户端兼容机器代码的二进制翻译,包括在所述本原客户端环境内与所述本原客户端环境被配置为在其上运行的所述主机硬件架构隔离地创建一组虚拟寄存器,所述一组虚拟寄存器表示由所述客户机软件在所述指定客户机硬件架构处执行时使用的寄存器,每个虚拟寄存器对于所述客户机软件的二进制翻译能够经由单个指令从所述本原客户端环境内访问,其中所述本原客户端环境被配置为在提供所述客户机软件到本原客户端机器代码的二进制翻译时执行所述本原客户端兼容机器代码,并且其中在不需要移植或者重新编译所述本原客户端兼容机器代码的情况下提供所述客户机软件到本原客户端机器代码的二进制翻译。

2. 根据权利要求1所述的方法,其中所述本原客户端环境访问在存储器中存储或者在主机处可用的寄存器中存储的一组模拟的客户机寄存器,其中与所述本原客户端兼容机器代码相关联的数据被存储在所述本原客户端环境内,以及其中所述模拟的客户机寄存器与所述指定客户机硬件架构的寄存器相对应。

3. 根据权利要求1所述的方法,其中所述虚拟寄存器的相应地址通过基址指针(RBP)加上预定偏移来引用。

4. 根据权利要求1所述的方法,其中,提供所述客户机软件到所述本原客户端兼容机器代码的二进制翻译包括:

在所述本原客户端环境内与所述本原客户端环境被配置为在其上运行的所述主机硬件架构隔离地模拟所述指定客户机硬件架构的特征和所述本原客户端环境内的所述客户机软件的应用编程接口(API)调用。

5. 根据权利要求1所述的方法,其中,提供所述客户机软件到所述本原客户端兼容机器代码的二进制翻译包括:

利用与所述本原客户端环境兼容的代码来替代所述客户机软件中的代码。

6. 根据权利要求1所述的方法,其中,所述客户机软件包括ARM软件或者x86软件。

7. 根据权利要求1所述的方法,其中,所述本原客户端兼容机器代码是安全的并且是可移植的,以及其中,所述客户机软件是不安全的或者不可移植的。

8. 根据权利要求1所述的方法,其中,所述客户机软件被配置为仅在特定客户机操作系统内执行,以及其中,所述本原客户端环境被配置为执行在多个不同操作系统中的任何一个操作系统内。

9. 根据权利要求1所述的方法,其中,所述本原客户端兼容机器代码能够在应用内执

行,并且其中所述应用是浏览器。

10. 一种非暂时性计算机可读介质,所述非暂时性计算机可读介质包括指令,所述指令在由一个或多个计算机执行时使所述一个或多个计算机实现一种方法,所述方法包括:

提供用于在主机硬件架构中运行本原客户端软件的本原客户端环境,其中在所述主机硬件架构上运行的主机软件对所述本原客户端软件不可访问;

接收要在所述本原客户端环境中运行的客户机软件,其中所述客户机软件被配置为执行在指定客户机硬件架构处而不是执行在所述主机硬件架构处或所述本原客户端环境内,以及其中所述指定客户机硬件架构不同于所述主机硬件架构和所述本原客户端环境;

在所述本原客户端环境内与所述本原客户端环境被配置为在其上运行的所述主机硬件架构隔离地生成在所述主机硬件架构中使用的硬件的第一组软件表示和在所述客户机硬件架构中使用的硬件的第二组软件表示;以及

使用模拟软件提供所述客户机软件到用于与所述本原客户端软件一起在所述本原客户端环境内执行并且不能访问在所述主机硬件架构上运行的所述主机软件的本原客户端兼容机器代码的二进制翻译,包括在所述本原客户端环境内与所述本原客户端环境被配置为在其上运行的所述主机硬件架构隔离地创建一组虚拟寄存器,所述一组虚拟寄存器表示由所述客户机软件在所述指定客户机硬件架构处执行时使用的寄存器,每个虚拟寄存器对于所述客户机软件的二进制翻译能够经由单个指令从所述本原客户端环境内访问,其中所述本原客户端环境被配置为在提供所述客户机软件到本原客户端机器代码的二进制翻译时执行所述本原客户端兼容机器代码,并且其中在不需要移植或者重新编译所述本原客户端兼容机器代码的情况下提供所述客户机软件到本原客户端机器代码的二进制翻译。

11. 根据权利要求10所述的非暂时性计算机可读介质,其中所述本原客户端环境访问在存储器中存储的或者在主机上可用的寄存器中存储的模拟的一组客户机寄存器,其中与所述本原客户端兼容机器代码相关联的数据被存储在所述本原客户端环境内,以及其中所述模拟的客户机寄存器与所述指定客户机硬件架构的寄存器相对应。

12. 根据权利要求10所述的非暂时性计算机可读介质,其中,提供所述客户机软件到所述本原客户端兼容机器代码的二进制翻译包括:

在所述本原客户端环境内与所述本原客户端环境被配置为在其上运行的所述主机硬件架构隔离地模拟所述指定客户机硬件架构的特征和所述本原客户端环境内的所述客户机软件的应用编程接口(API)调用。

13. 根据权利要求10所述的非暂时性计算机可读介质,其中,提供所述客户机软件到所述本原客户端兼容机器代码的二进制翻译包括:

利用与所述本原客户端环境兼容的代码来替代所述客户机软件中的代码。

14. 根据权利要求10所述的非暂时性计算机可读介质,其中,所述客户机软件包括ARM软件或者x86软件。

15. 根据权利要求10所述的非暂时性计算机可读介质,其中,所述本原客户端兼容机器代码是安全的并且是可移植的,以及其中,所述客户机软件是不安全的或者不可移植的。

16. 一种用于二进制翻译的系统,包括:

一个或多个处理器;以及

存储器,所述存储器包括指令,所述指令在由所述一个或多个处理器执行时使所述一

个或多个处理器实现一种方法,所述方法包括:

提供用于在主机硬件架构中运行本原客户端软件的本原客户端环境,其中在所述主机硬件架构上运行的主机软件对所述本原客户端软件不可访问;

接收要在所述本原客户端环境中运行的客户机软件,其中所述客户机软件被配置为执行在指定客户机硬件架构处而不是执行在所述主机硬件架构处或所述本原客户端环境内,以及其中所述指定客户机硬件架构不同于所述主机硬件架构和所述本原客户端环境;

在所述本原客户端环境内与所述本原客户端环境被配置为在其上运行的所述主机硬件架构隔离地生成在所述主机硬件架构中使用的硬件的第一组软件表示和在所述客户机硬件架构中使用的硬件的第二组软件表示;以及

使用模拟软件提供所述客户机软件到用于与所述本原客户端软件一起在所述本原客户端环境内执行并且不能访问在所述主机硬件架构上运行的所述主机软件的本原客户端兼容机器代码的二进制翻译,包括在所述本原客户端环境内与所述本原客户端环境被配置为在其上运行的所述主机硬件架构隔离地创建一组虚拟寄存器,所述一组虚拟寄存器表示由所述客户机软件在所述指定客户机硬件架构处执行时使用的寄存器,每个虚拟寄存器对于所述客户机软件的二进制翻译能够经由单个指令从所述本原客户端环境内访问,其中所述本原客户端环境被配置为在提供所述客户机软件到本原客户端机器代码的二进制翻译时执行所述本原客户端兼容机器代码,并且其中在不需要移植或者重新编译所述本原客户端兼容机器代码的情况下提供所述客户机软件到本原客户端机器代码的二进制翻译。

17. 根据权利要求16所述的系统,其中所述虚拟寄存器的相应地址通过基址指针(RBP)加上预定偏移来引用。

18. 根据权利要求16所述的系统,其中,提供所述客户机软件到所述本原客户端兼容机器代码的二进制翻译包括:

在所述本原客户端环境内与所述本原客户端环境被配置为在其上运行的所述主机硬件架构隔离地模拟所述指定客户机硬件架构的特征和所述本原客户端环境内的所述客户机软件的应用编程接口(API)调用。

19. 根据权利要求16所述的系统,其中,提供所述客户机软件到所述本原客户端兼容机器代码的二进制翻译包括:

利用与所述本原客户端环境兼容的代码来替代所述客户机软件中的代码。

20. 根据权利要求16所述的系统,其中,所述客户机软件包括ARM软件或者x86软件。

用于到本原客户端的二进制翻译的方法、系统和介质

[0001] 相关申请的交叉引用

[0002] 本申请涉及同时提交的具有代理人案号096553-0073并且标题为“对共享对象级的二进制翻译(BINARY TRANSLATION ON SHARED OBJECT LEVEL)”的国际专利申请,其公开内容以引用的方式全部并入在本文中。

背景技术

[0003] 本主题技术大体上涉及二进制翻译技术。一些软件被编译为用于特定CPU架构(诸如, ARM®或者x86®)和特定操作系统(诸如, Android®或者Microsoft Windows®)的二进制程序。可以通过互联网将二进制程序下载至用户的计算机。然而,用户可能不信任该程序并且可能希望在其中该程序具有对在该程序外部存储的计算机上的数据的有限访问的安全模式下运行该程序。如上文说明的是,可能需要一种用于计算机上安全地执行软件的方法。

[0004] 本原客户端(Native Client)可以被用于安全地执行计算机上的软件。然而,一些软件不是为本原客户端编写的并且与本原客户端不兼容。可能需要重新编译这种软件并且将这种软件移植到本原客户端,对于一些大型的现代化软件产品,这可能需要付出巨大努力。如上文所说明,可能需要将与本原客户端不兼容的代码翻成本原客户端。

发明内容

[0005] 根据一些方面,本主题技术涉及一种方法。该方法包括:接收要在本原客户端环境中运行的客户机软件,其中,客户机软件被配置为执行在指定客户机硬件架构处而不是执行在本原客户端环境内。该方法包括:使用模拟软件提供客户机软件到本原客户端兼容机器代码的二进制翻译,其中,该本原客户端兼容机器代码在用于本原客户端环境的沙盒内执行,其中,该本原客户端兼容机器代码能够在应用内执行,其中,提供客户机软件到用于在沙盒内执行的本原客户端兼容机器代码的二进制翻译刚好及时发生在模拟的客户机软件的运行时间期间,并且不需要移植或者重新编译客户机软件,以及其中,提供二进制翻译与执行模拟的客户机软件交错进行。该方法包括:在本原客户端环境中和在沙盒内运行本原客户端兼容机器代码。

[0006] 根据一些方面,本主题技术涉及一种非暂时性计算机可读存储介质,该非暂时性计算机可读存储介质包括指令。该指令包括用于接收要在本原客户端环境中运行的客户机软件的代码,其中,该客户机软件被配置为执行在指定客户机硬件架构处而不是执行在本原客户端环境内。该指令包括用于使用模拟软件提供客户机软件到本原客户端兼容机器代码的代码的二进制翻译,其中,该本原客户端兼容机器代码在用于本原客户端兼容机器代码的沙盒内执行,其中,该本原客户端兼容机器代码能够在应用内执行,其中,提供客户机软件到在沙盒内执行的本原客户端兼容机器代码的二进制翻译刚好及时发生在模拟的客户机软件的运行时间期间,并且不需要移植或者重新编译客户机软件,以及其中,提供二进制翻译与执行模拟的客户机软件交错进行。提供客户机软件到本原客户端兼容机器代码的

二进制翻译包括：创建虚拟寄存器的集合，该虚拟寄存器的集合表示由客户机软件在指定客户机硬件架构处执行时使用的寄存器，其中，虚拟寄存器的所述地址被基址指针 (RBP) 加上预定偏移加以引用，以及其中，在该虚拟寄存器的集合中的各个虚拟寄存器是能够从沙盒内经由单个指令而访问。

[0007] 根据一些方面，本主题技术涉及一种系统。该系统包括一个或多个处理器和存储指令的存储器。该指令包括用于接收要在本原客户端环境中运行的客户机软件的代码，其中，客户机软件被配置为执行在指定客户机硬件架构处而不是执行在本原客户端环境内。该指令包括用于使用模拟软件提供客户机软件到本原客户端兼容机器代码的代码的二进制翻译，其中，该本原客户端兼容机器代码在本原客户端环境的沙盒内执行，其中，本原客户端兼容机器代码能够在应用内执行，其中，提供客户机软件到用于在沙盒内执行的本原客户端的二进制翻译刚好及时发生在模拟的客户机软件的运行时间期间，并且不需要移植或者重新编译客户机软件，其中，用于本原客户端环境的沙盒访问在存储器中存储或者在主机处可用的寄存器中存储的模拟的客户机寄存器的集合，其中，与本原客户端兼容机器代码相关联的数据被存储在用于本原客户端环境的沙盒内，以及其中，模拟的客户机寄存器与指定客户机硬件架构的寄存器相对应。

[0008] 要理解的是，本主题技术的其它配置将会通过以下的详细说明而变得显而易见，在以下详细说明中，通过说明的方式示出并且描述了本主题技术的各种配置。如将会认识到的是，在不脱离本主题技术的范围的情况下，本主题技术能够具有其它配置和不同配置并且能够对其若干细节在各个其它方面进行修改。因此，在本质上应当将附图和详细说明视作是说明性的，而不是限制性的。

附图说明

[0009] 在随附权利要求书中阐述了本主题技术的特征。然而，出于解释之目的，在以下的图中阐述了所公开的主题的若干方面。

[0010] 图1A图示了可以涉及到本原客户端的二进制翻译的示例客户机。

[0011] 图1B图示了可以涉及到本原客户端的二进制翻译的示例主机。

[0012] 图2图示了可以通过其完成到本原客户端的二进制翻译的示例过程。

[0013] 图3概念性地图示了具有实现本主题技术的一些实现方式的示例电子系统。

具体实施方式

[0014] 以下阐述的详细说明旨在作为对本主题技术的各种配置的描述，并且不旨在表示可以实践本主题技术的唯一配置。将附图并入本文中并且构成详细说明的一部分。详细说明包括具体细节，用于提供对本主题技术的全面理解。然而，清楚且显而易见的是，本主题技术不限于本文中阐述的具体细节并且可以在没有这些具体细节的情况下实践。在某些情况下，以框图形式示出了某些结构和部件，以便避免模糊本主题的构思。

[0015] 如本文所使用的，“本原客户端”可以是指部署软件故障隔离和二进制代码验证方法的沙盒化环境。本原客户端实现与可移植式操作系统接口 (POSIX) 类似的应用编程接口 (API) 的有限的集合并且能够在web浏览器内执行。“二进制翻译”可以是指一种分析在一个平台 (客户机) 上执行的应用或者操作系统的机器代码 (称为客户机代码) 以产生适合于在

不同平台(主机)上执行的代码的机制。“平台”可以包括硬件和软件栈组合。应用通常是针对特定平台(诸如,ARM或者x86)开发的。上述术语中的每个还涵盖其简单且普通的含义。

[0016] 本主题技术涉及在拥有更好的API级可移植性、安全性、和可靠性特性的另一平台(主机平台(例如,本原客户端x86-64))上运行针对客户机平台(例如,Android ARM)开发的现有软件的二进制翻译技术。可以利用可以以动态加载库的形式访问本原部件的移植性编程语言(例如,Java)来写入用于客户机平台的应用。可以将主机部件称为共享对象,并且可以使用客户机平台的硬件架构和操作系统、在客户机平台处编译该主机部件。为了支持应用,主机平台可以需要运行Java和主机代码二者。本主题技术涉及在本原客户端沙盒内运行非移植性的、客户机平台专用应用库。

[0017] 根据本主题技术的一些实现方式,为了在本原客户端沙盒内执行客户机代码,产生符合本原客户端沙盒化规则的主机代码。为了实现合理的性能,可以应用优化技术。该优化技术可以包括:产生模拟客户机代码的行为的即时软件故障隔离(SFI)兼容主机代码。该优化技术可以包括:将模拟的客户机寄存器存储在快速存储件(诸如,在主机寄存器中或者在基址指针(RBP)相对存储器中的缓存器)中。这个主机代码的产生在不需要对模拟的客户机情境进行明确地沙盒化访问的情况下而允许快速访问。该优化技术可以包括:使用主机的共享对象来模拟对外部平台共享对象的调用,换言之,若可能,对标准库进行前述二进制翻译。在同时提交的具有代理人案号093054-0893并且标题为“对共享对象级的二进制翻译(BINARY TRANSLATION ON SHARED OBJECT LEVEL)”的美国专利申请(其公开内容以引用的方式全部并入在本文中)中详细描述了该模拟策略。

[0018] 在一些情况下,例如,由于安全性要求,主机平台可以不具有对在客户机平台处可用的一些API访问的权限。使用附加技术以模拟非可用功能。

[0019] 一些中央处理单元(CPU)架构经由页面表中的适当位(X位)支持对可执行代码的显式标记。来自虚拟存储器的代码可以仅在设置了X位的条件下执行。在一些情况下,出于安全性原因不允许在本原客户端沙盒内直接支持X位操纵。另外,由于通过二进制翻译器的客户机代码执行未考虑到客户机代码上存在主机X位,所以这种操纵可能无法达到期望的效果。为了模拟与X位的操纵有关的功能(例如,通常使用POSIX系统上的mmap(PROT_EXEC) API来提供),通过二进制翻译引擎不需要考虑由主机mmap() API操纵的附加结构。这可以由存储器中的存储特定客户机页面是可执行还是不可执行(例如,特定客户机页面是否具有其X位集合)的位图来完成。为了避免在执行每一个指令时检查位图,可以在翻译时间期间发生X位检查。稍后,改变用于可执行页面的X位可以导致代码缓存刷新。

[0020] 本原客户端可以不支持信号的POSIX特征。信号可以被用于提供关于应用中的异常情况(诸如,死机或者执行失败)的信息。信号还可以被用于垃圾收集和管理语言运行时间。在一些情况下,除非二进制翻译引擎模拟信号处理程序的设置并且将信号递送至客户机专用的信号处理程序,否则使用信号的软件可能无法在本原客户端沙盒中运行。因此,为了模拟信号,二进制翻译引擎可以维持有源信号掩码。如果要递送信号,则可以将本原客户端内执行二进制翻译的软件暂停,并且可以执行信号处理程序。在已经执行了信号处理程序之后,可以恢复二进制翻译的软件的执行。这个信号处理可以通过添加对二进制翻译调度器内的信号掩码的检查并且使用函数包装机制来明确地调用具有适当参数的信号处理程序来实现。

[0021] 本原客户端是一种用于在浏览器中安全地运行本原编译代码(例如,在一些情况下,编译的C或者C++代码)的沙盒化技术。如本文中使用的是,短语“沙盒”或者“沙盒化技术”可以是指用于通过限制程序可以访问的存储器中的空间来分离运行程序的计算机安全机制。沙盒程序被称为能够对“沙盒内”的存储器区域访问并无法对“沙盒外”的存储器区域访问。本原客户端代码独立于操作系统,并且只要对操作系统实现支持,就能够在任何操作系统上运行。本原客户端允许程序员通过web浏览器来开发、分配、和执行任何客户端系统的本原代码,同时维持浏览器的安全性特征。本原客户端提供执行沙盒内的本原代码的能力,并且不允许对浏览器或者对沙盒外的主机操作系统直接访问。本原客户端可以通过受控制的API提供对浏览器或者对主机操作系统的有限访问。对本原客户端兼容机器代码的二进制翻译确保仅发生合法的沙盒访问。虽然本文中结合浏览器描述了本主题技术的各个方面,但是在可替代实现方式中,可以使用另一应用来替代浏览器。

[0022] 本主题技术将与本原客户端不兼容的软件翻成本原客户端兼容机器代码。有利地,用户可以从他/她不信任的源运行软件,并且可以确保软件将会在沙盒内运行并且将会具有对在用户的在沙盒外的计算机上存储的信息的有限访问。在本原客户端内运行的模拟的客户机软件的运行时间期间发生翻译。在执行代码时,在代码的运行时间期间发生“即时”翻译。结果是,仅对被执行的并且需要翻译的代码进行翻译。不对代码的未执行的部分进行翻译,从而节省时间和处理资源。可以连同有价值的(nontrivial)移植层部署二进制翻译技术。二进制翻译器可以对客户机软件的代码进行翻译。然而,可以使用在模拟软件中设置的移植层执行标准库调用。移植层可以将客户机平台的预期API翻成本原客户端上的可用API的集合。在一些情况下,这种翻译可以包括:将同步阻塞库调用翻译成对本原客户端的异步非阻塞调用,以及在本原客户端内模拟本原客户端中当前缺少的客户机硬件的特征。

[0023] 二进制翻译可以将客户机软件翻译成适合于进一步验证和沙盒化的格式。软件故障隔离(SFI)技术可以被用于实现更好的可移植性和安全性。

[0024] 本主题技术可以被视为二进制翻译的特殊情况,其中,本原客户端是主机的操作系统,从而产生能够在任何操作系统内执行的浏览器中运行的安全程序。由于翻成本原客户端,所以增加软件的安全性和可移植性二者。

[0025] 在二进制翻译期间,可以将代码翻译成与本原客户端的规则兼容的格式。例如,本原客户端要求不提供跨越是32的倍数的地址的指令(换言之,指令是管束对齐的)。结果是,当二进制翻译器发出机器代码时,二进制翻译器确保不会跨越管束边界(32的倍数)。

[0026] 在本主题技术的一些方面中,计算机接收要在本原客户端环境中运行的客户机软件。该客户机软件被配置为在指定客户机硬件架构(例如,x86硬件或者ARM硬件)处执行。该客户机软件可以不是安全的或者可以不是可移植的。计算机将该客户机软件二进制翻成本原客户端兼容机器代码。本原客户端兼容机器代码在限制本原客户端兼容机器代码的对存储在主机处的数据的访问的沙盒内执行。本原客户端兼容机器代码是安全的、可移植的、和可在支持浏览器的任何操作系统中的浏览器内执行。

[0027] 沙盒可以能够对主机上可用的寄存器的集合中的子集访问。可以经由寄存器的集合中的子集或者经由定位在距该子集中的存储器的一定偏移量处的存储器将重要客户机数据(诸如,客户机寄存器)从客户机软件提供给本原客户端兼容机器代码。由于二进制

翻译的代码可以需要密集使用客户机寄存器,所以该方法可以允许高性能地模拟客户机硬件的同时,仍然满足本原客户端沙盒的要求。

[0028] 提供客户机软件到本原客户端兼容机器代码的二进制翻译可以包括:创建虚拟寄存器的集合,该虚拟寄存器的集合表示由客户机软件在主机硬件架构处执行时使用的寄存器。虚拟寄存器的地址可以被RBP加上预定偏移加以引用。例如,寄存器R0可以在位置RBP处,寄存器R1可以在位置RBP+4个偏移字节处,寄存器R2可以在位置RBP+8个偏移字节处等。结果是,在虚拟寄存器的集合中的每个虚拟寄存器可以是可从沙盒内经由单个指令而访问。例如,如果写入命令具有以下格式:WRITE (POSITION, VALUE),则将值“0”写入到寄存器R2的命令可以被写入为:WRITE (RBP+8, 0)。因为这个技术导致所生成的代码的最大性能,所以单个指令可以是有利的。本原客户端沙盒化机制允许针对RBP相对寻址的特定异常。具体地,能够在二进制翻译的代码中使用单个[RBP+4*N]指令以对寄存器编号N访问。RBP寄存器被使用作为基址。在一些情况下,可以需要附加沙盒化指令以确存储器访问处于沙盒内。

[0029] 提供客户机软件到本原客户端兼容机器代码的二进制翻译可以包括:模拟指定客户机硬件架构的特征和本原客户端内的客户机软件的应用编程接口(API)调用。指定客户机硬件架构的特征可以包括寄存器、栈、指针等。API调用可以包括对系统API或者与客户机软件相关联的API的调用。寄存器、栈、指针等被模拟,这意味着本原客户端兼容机器代码创建硬件项的软件表示,该硬件项的软件表示与硬件项的行为相似,但是存在于在处理器和存储器(不具有相关联的硬件部件的情况下)上执行的软件中。访问模拟的部件的调用可以与访问客户机上的物理部件的调用完全相同或者相似,从而允许在模拟的环境中运行未修改的客户机代码。

[0030] 为了模拟客户机架构的在本原客户端环境内不可用的某些方面(诸如,可执行位(X位)和异步信号处理),可以使用附加软件模拟逻辑。例如,本原客户端不允许经由mmap (PROT_EXEC)接口对可执行代码区域或者可用于POSIX系统的代码区域进行自由的控制。因此,模拟软件可以经由附加机制实现对可执行位的控制。模拟软件可以经由单独的数据结构来保持对可执行客户机代码追踪。类似地,当本原客户端不支持信号递送时,模拟软件可以通过周期性地检查待定信号掩码并且一旦信号升高就明确地调用信号处理程序来支持发送异步信号。如本文所使用的是,短语“模拟软件”或者“模拟器”可以是指允许客户机软件在本原客户端环境内运行的软件。

[0031] 图1A图示了可以涉及本原客户端到二进制翻译的示例客户机100A。客户机100A可以是任何计算装置,例如,膝上型计算机、台式计算机、平板计算机、移动电话、个人数字助理(PDA)、电子音乐播放器、智能手表、与一个或多个处理器和存储器耦合的电视机等。在一些示例中,客户机100A具有ARM硬件。可替代地,客户机100A可以具有x86硬件。

[0032] 如图所示,客户机100A包括处理单元102A、网络接口104A、和存储器106A。处理单元102A包括一个或多个处理器。处理单元102A可以包括中央处理单元(CPU)、图形处理单元(GPU)、或者任何其它处理单元。处理单元102A执行在计算机可读介质(例如,存储器106A)中存储的计算机指令。网络接口104A允许客户机100A在网络(例如,互联网、内联网、蜂窝网络、局域网、广域网、有线网络、无线网络、虚拟专用网络(VPN)等)中传输和接收数据。存储器106A存储数据和/或指令。存储器106A可以是缓存单元、存储单元、内部存储器单元、或者外部存储器单元中的一个或多个。如图所图示,存储器106A包括客户机寄存器108A、客户机

软件程序110A、和客户机API 112A。

[0033] 客户机寄存器108A是客户机100A上的与客户机100A的硬件架构(例如,ARM或者x86)相关联的寄存器。除了寄存器108A之外,存储器106A可以包括诸如栈或者指针的其它硬件架构。客户机API 112A是客户机100A上存在的API,可以使用该API来将用于客户机100A及其相关联的硬件架构的软件写入。客户机API 112A可以包括与客户机100A的硬件相关联的系统API或者由客户机软件程序110A的供应商连同客户机软件程序110A一起提供的供应商API。

[0034] 客户机软件程序110A是被配置为在客户机100A的硬件架构上执行并且被配置为与客户机寄存器108A和与客户机API 112A接口连接(interface)的软件程序。客户机软件程序110A可以不能够在本原客户端环境中或者在与客户机100A的硬件架构不同的硬件架构上执行。例如,如果客户机100A具有ARM硬件,则客户机软件程序110A可以被配置为在ARM硬件上但不能在x86硬件上执行。

[0035] 图1B图示了可以涉及对共享对象级二进制翻译的示例主机100B。主机100B可以是任何计算装置,例如,膝上型计算机、台式计算机、平板计算机、移动电话、个人数字助理(PDA)、电子音乐播放器、智能手表、与一个或多个处理器和存储器耦合的电视机等。主机100B具有与客户机100A的硬件架构不同的硬件架构。例如,如果客户机100A的硬件架构是ARM硬件,则主机100B的硬件架构不是ARM硬件,并且例如,可以是x86硬件。可替代地,主机100B可以具有与客户机100A相同的硬件架构。

[0036] 如图所示,主机100B包括处理单元102B、网络接口104B、和存储器106B。处理单元102B包括一个或多个处理器。处理单元102B可以包括中央处理单元(CPU)、图形处理单元(GPU)、或者任何其它处理单元。处理单元102B执行在计算机可读介质(例如,存储器106B)中存储的计算机指令。网络接口104B允许客户机100B在网络(例如,互联网、内联网、蜂窝网络、局域网、广域网、有线网络、无线网络、虚拟专用网络(VPN)等)中传输和接收数据。存储器106B存储数据和/或指令。存储器106B可以是缓存单元、存储单元、内部存储器单元、或者外部存储器单元中的一个或多个。如图所图示,存储器106B包括安全的主机寄存器114B、安全的主机内容116B、和本原客户端沙盒120B。

[0037] 安全的主机寄存器114B是主机100B上的与客户机100B的硬件架构(例如,ARM或者x86)相关联的寄存器。除了安全的主机寄存器114B之外,存储器106B可以包括诸如栈或者指针的其它硬件架构。安全的主机内容116B包括存储在主机100B上的诸如文字处理文档、照片、视频、音频、文件等的内容。安全的主机寄存器114B和安全的主机内容116B在本原客户端沙盒120B外并且不可从本原客户端沙盒120B访问。

[0038] 本原客户端沙盒120B是其中能够执行本原客户端代码的存储器106B的安全区域。本原客户端代码可以在本原客户端沙盒120B内执行,并且不可以对本原客户端沙盒120B外的寄存器(诸如,安全的主机寄存器114B或者安全的主机内容116B)或者数据访问。以这种方式,主机100B的用户可以在确认他/她的个人数据和主机100B的安全寄存器对潜在地非可信代码是不可访问的时执行潜在地非可信本原客户端代码。

[0039] 如图所示,本原客户端沙盒120B包括可访问的主机寄存器122B、模拟的客户机寄存器108B、本原客户端兼容机器代码110B、和模拟的客户机API 112B。本原客户端兼容机器代码110B与本原客户端到二进制翻译的客户机软件程序110A相对应。本原客户端兼容机器

代码1110B在本原客户端沙盒120B内执行。与其它本原客户端代码相似,本原客户端兼容机器代码110B不可以对本原客户端沙盒120B外的寄存器(诸如,安全的主机寄存器114B或者安全的主机内容116B)或者数据访问。本原客户端兼容机器代码110B在基本上任何硬件或者操作系统上的浏览器内是可执行的。具体地,本原客户端兼容机器代码110B可以在主机100B上执行,该主机100B可以具有与客户机100A的硬件架构不同的硬件架构。在一些情况下,客户机软件程序110A到本原客户端兼容机器代码的二进制翻译可以刚好及时发生在本原客户端兼容机器代码程序的运行时间期间,并且不需要移植或者重新编译本原客户端兼容机器代码。结果是,仅对被执行的并且需要翻译的代码进行翻译。不对代码的未执行的部分进行翻译,从而节省时间和处理资源。

[0040] 模拟的客户机寄存器108B与在本原客户端沙盒120B内执行的软件中模拟的客户机100A的客户机寄存器108A相对应。还可以在本原客户端沙盒120B中模拟由客户机100A上执行的客户机软件程序110A使用的诸如栈或者指针的其它硬件。寄存器108B被模拟意味着本原客户端兼容机器代码创建硬件项的软件表示,该硬件项的软件表示与硬件项的行为相似,但是存在于在处理器和存储器(不具有相关联的硬件部件的情况下)上执行的软件中。访问模拟的部件的调用可以与访问客户机上的物理部件的调用完全相同或者相似,从而实现容易地允许将代码从客户机100A移植到本原客户端沙盒120B。根据一些情况,用与本原客户端环境兼容的代码来替代客户机软件程序110A的代码。

[0041] 模拟的客户机API 112B与客户机100A的在本原客户端沙盒120B内执行的软件中执行二进制翻译或者模拟的客户机API 112A相对应。模拟的客户机API 112B可以包括对与客户机100A的硬件相关联的系统API或者由客户机软件程序110A的供应商连同客户机软件程序110A一起提供的供应商API的模拟。与客户机软件程序110A对客户机100A处的客户机寄存器108A和客户机API 112A访问的方式相似,对应的本原客户端兼容机器代码110B可以对本原客户端沙盒120B处的模拟的客户机寄存器108B和模拟的客户机API 112B访问。

[0042] 本原客户端沙盒120B还包括可访问的主机寄存器122B。可访问的主机寄存器122B是主机100B的可由在本原客户端沙盒120B内执行的软件(包括本原客户端兼容机器代码110B)、从本原客户端沙盒120B内访问的寄存器。主机的寄存器可以是对本原客户端兼容机器代码不可访问的安全的主机寄存器114B或者是可以由本原客户端兼容机器代码访问的可访问的主机寄存器122B。

[0043] 图2图示了可以通过其完成到本原客户端的二进制翻译的示例过程200。

[0044] 处理器200从步骤210处开始,在步骤210中主机(例如,主机100B)接收以在本原客户端环境(例如,本原客户端沙盒120B)中运行的客户机软件(例如,客户机软件程序110A)。客户机软件被配置为在指定客户机硬件架构(例如,客户机100A)处而不是在本原客户端环境内执行。本原客户端环境被配置为在多个不同的硬件架构中的任何一个内执行。例如,本原客户端环境可以被配置为在可以存在于多个不同的硬件架构中的任何一个中的应用(诸如,浏览器)内执行。

[0045] 在步骤220中,主机使用模拟软件提供客户机软件到本原客户端兼容机器代码(例如,本原客户端兼容机器代码110B)的二进制翻译。本原客户端兼容机器代码在用于本原客户端环境的沙盒(例如,本原客户端沙盒120B)内执行。本原客户端兼容机器代码可在应用(诸如,浏览器)内执行。提供客户机软件到用于在沙盒内执行的本原客户端兼容机器代码

的二进制翻译刚好及时发生在模拟的客户机软件的运行时间期间,并且不需要移植或者重新编译客户机软件。提供二进制翻译与执行模拟的客户机软件交错进行。

[0046] 客户机软件可以包括例如设计为在ARM硬件系统或者x86硬件系统上执行的ARM软件或者x86软件。可以利用本原客户端代码或者与本原客户端环境兼容的代码来替代客户机软件中的一些代码。在一些情况下,客户机软件可以是不安全的或者是不可移植的。本原客户端兼容机器代码可以是安全并且是可移植的。客户机软件可以被配置为仅在指定操作系统内执行。本原客户端环境可以能够在多个操作系统中的任何一个内执行。

[0047] 提供客户机软件到本原客户端兼容机器代码的二进制翻译可以包括:创建表示由客户机软件在指定客户机硬件架构处执行时使用的寄存器(例如,客户机寄存器108A)的虚拟寄存器的集合(例如,模拟的客户机寄存器108B)。虚拟寄存器的地址可以被基址指针(RBP)加上预定偏移加以引用。由于RBP加上偏移技术,各个寄存器可以从沙盒内经由单个指令访问。例如,寄存器R0可以在地址RBP处,寄存器R1可以在地址RBP+4处,寄存器R2可以在地址RBP+8处,寄存器Rn可以在地址RBP+4n处等(其中,n是整数)。

[0048] 提供客户机软件到本原客户端兼容机器代码的二进制翻译可以包括:模拟客户机硬件架构的特征和本原客户端内的客户机软件的API调用。可以二进制翻译来自客户机硬件架构的系统API或者供应商API以在本原客户端沙盒内执行。可替代地,可以利用本原客户端沙盒的系统API替代客户机硬件架构的系统API中的一些。

[0049] 在二进制翻译期间,可以将客户机软件翻译成与本原客户端的规则兼容的格式。例如,本原客户端要求不提供跨越可被32整除的地址的指令。在这种情况下,可以将可被32整除的地址称为“管束边界(bundle boundary)”。将由模拟器产生的所有指令以如下这种方式发出:指令不会越过管束边界并且利用无操作(NOP)指令进行填充。用于本原客户端环境的沙盒访问在存储器中或者在主机处的可用的寄存器中存储的模拟的客户机寄存器的集合。与本原客户端兼容机器代码相关联的数据被存储在用于本原客户端环境的沙盒内。模拟的客户机寄存器与指定客户机硬件架构的寄存器相对应。

[0050] 在步骤230中,主机在本原客户端环境中和在沙盒内运行本原客户端兼容机器代码。本原客户端兼容机器代码可以对本原客户端沙盒内的寄存器和数据访问,但不可以对本原客户端沙盒外的寄存器或者数据访问。沙盒化软件对在主机的寄存器中或者存储在存储器中暂时存储的模拟的客户机寄存器访问。在步骤230之后,过程200结束。

[0051] 如上所述,根据特定顺序并且连续地实现过程200的步骤210-230。然而,可以以任何顺序实现步骤210-230。在一些示例中,可以并行实现步骤210-230中的两个或更多个。

[0052] 图3概念性地图示了实现本主题技术的一些实现方式的电子系统300。例如,可以使用电子系统300的布置来实现客户机100A和主机100B中的一个或多个。电子系统300能够是计算机(例如,移动电话、PDA),或者任何其它种类电子装置。这种电子系统包括各种类型的计算机可读介质和用于各种其它类型的计算机可读介质的接口。电子系统300包括总线305、处理器310、系统存储器315、只读存储器320、永久性存储装置325、输入装置接口330、输出装置接口335、和网络接口340。

[0053] 总线305共同地表示通信地连接电子系统300的若干内部装置的所有系统、外围装置、和芯片集总线。例如,总线305将处理器310与只读存储器320、系统存储器315、和永久性存储装置325通信地连接。

[0054] 处理器310从这些各种存储器单元检索要执行的指令和要处理的数据,以便执行本主题技术的处理。在不同实现方式中,处理器能够包括单个处理器或者多核处理器。

[0055] 只读存储器 (ROM) 320存储由电子系统的处理器310和其它模块需要的静态数据和指令。另一方面,永久性存储装置325是读写存储器装置。该装置是非易失性存储器单元,该非易失性存储器单元即使在电子系统300关闭时也存储指令和数据。本主题技术的一些实现方式使用大容量存储装置 (例如,磁盘或者光盘及其相对应的磁盘驱动器) 作为永久性存储装置325。

[0056] 其它实现方式使用可移动存储装置 (例如,软盘、闪速驱动器、或者磁盘驱动器) 作为永久性存储装置325。与永久性存储装置325相同的是,系统存储器315是读写存储器装置。然而,与存储装置325不同的是,系统存储器315是诸如随机存取存储器的易失性读写存储器。系统存储器315存储处理器在运行时间处需要的指令和数据中的一些。在一些实现方式中,本主题技术的过程被存储在系统存储器315、永久性存储装置325、或者只读存储器320中。例如,各种存储器单元包括根据一些实现方式的用于到本原客户端的二进制翻译的指令。处理器310从这些各种存储器单元检索要执行的指令和要处理的数据,以便执行一些实现方式的过程。

[0057] 总线305还连接至输入装置接口330和输出装置接口335。输入装置接口330使用户能够向电子系统传送信息和选择命令。与输入装置接口330一起使用的输入装置包括,例如,字母数字键盘和指向装置 (还称为“光标控制装置”)。输出装置接口335使能够显示例如由电子系统300生成的图像。与输出装置接口335一起使用的输出装置包括,例如打印机和显示装置、例如阴极射线管 (CRT) 或者液晶显示器 (LCD)。一些实现方式包括例如用作输入装置和输出装置二者的触摸屏的装置。

[0058] 最后,如图3所示,总线305还通过网络接口340将电子系统300耦合至网络 (未示出)。以这种方式,电子系统300能够是计算机的网络 (例如,局域网 (LAN)、广域网 (WAN)、或内联网、或者例如互联网的一个或多个网络) 的部分。电子系统300的任何或者所有部件能够与结合本主题技术一起使用。

[0059] 上述特征和应用能够被实现为软件过程,该软件过程被指定为在计算机可读存储介质 (还称为计算机可读介质) 上记录的指令的集合。当这些指令由一个或多个处理器 (该一个或多个处理器可以包括例如一个或多个处理器、处理器的核、或者其它处理单元) 执行时,这些指令使处理器执行指令中指示的动作。计算机可读介质的示例包括,但不限于,CD-ROM、闪速驱动器、RAM芯片、硬盘驱动器、EPROM等。计算机可读介质不包括无线地或者通过有线连接传递的载波和电子信号。

[0060] 在本说明书中,术语“软件”意味着包括在只读存储器中驻留的固件或者在磁存储装置或者闪速存储装置 (例如,固态驱动器) 中存储的应用,其能够被读入到用于由处理器处理的存储器中。另外,在一些实现方式中,多种软件技术能够被实现为较大程序的子部分的同时,维持不同的软件技术。在一些实现方式中,多种软件技术还能够被实现为单独的程序。最后,共同实现本文中描述的软件技术的单独的程序的任何组合是在本主题技术的范围内。在一些实现方式中,软件程序在被安装为运行在一个或多个电子系统上时定义出执行和施行软件程序的操作的一个或多个特定的机器实现方式。

[0061] 能够以任何形式的编程语言 (包括编译语言或者解释语言、声明性语言或者过程

性语言)来写入计算机程序(还称为程序、软件、软件应用、脚本、或者代码),并且能够以任何形式(包括作为适合于计算环境中使用的独立式程序或者模块、部件、子例程、对象、或者其它单元)来部署该计算机程序。计算机程序可以但并非必须与文件系统中的文件相对应。能够将程序存储在保持其它程序或者数据(例如,在标记语言文档中存储的一个或多个脚本)的文件的部分中,或者存储在专用于所探讨中的程序的单个文件中,或者存储在多个协同文件(例如,存储一个或多个模块、子程序、或者代码的部分的文件)中。能够将计算机程序部署为在一个计算机上执行,或者在位于一个站点处或者跨域在多个站点分布的并且通过通信网络互相连接的多个计算机上执行。

[0062] 能够在数字电子电路系统、计算机软件、固件、或者硬件中实现以上描述的这些功能。能够使用一个或多个计算机程序产品来实现该技术。可编程处理器和计算机能够被包括在移动装置中或者封装为移动装置。能够通过一个或多个可编程处理器或者通过一个或多个可编程逻辑电路系统来执行过程和逻辑流程。能够通过通信网络将通用计算装置和专用计算装置与存储装置互相连接。

[0063] 一些实现方式包括电子部件(例如,微处理器、存储件、和存储器),该电子部件将计算机程序指令存储在机器可读或者计算机可读介质(可替代地,称为计算机可读存储介质、机器可读介质、或者机器可读存储介质)中。这种计算机可读介质的一些示例包括RAM、ROM、只读光盘(CD-ROM)、可记录光盘(CD-R)、可重写光盘(CD-RW)、只读数字通用光盘(例如,DVD-ROM、双层DVD-ROM)、各种可记录/可重写DVD(例如,DVD-RAM、DVD-RW、DVD+RW等)、闪存存储器(例如,SD卡、迷你SD卡、微型SD卡等)、磁性或者固态硬盘驱动器、只读和可记录Blu-Ray®光盘、超密度光盘、任何其它光学或者磁性介质、以及软盘。计算机可读介质能够存储计算机程序,该计算机程序可由至少一个处理器执行并且包括用于执行各种操作的指令的集合。计算机程序或者计算机代码的示例包括例如由编译器产生的机器代码,以及包括由计算机、电子部件、或者微处理器,使用解释器执行的高级代码的文件。

[0064] 虽然以上的讨论主要涉及执行软件的微处理器或者多核处理器,但是一些实现方式由一个或多个集成电路(例如,专用集成电路(ASIC)或者现场可编程门阵列(FPGA))执行。在一些实现方式中,这种集成电路执行在电路本身上存储的指令。

[0065] 如本说明书和本申请的任何权利要求项使用的是,术语“计算机”、“服务器”、“处理器”、和“存储器”全部是指电子装置或者其它技术装置。这些术语不包括人或者人群。针对本说明书的目的,术语显示(display)或者显示(displaying)意味着在电子装置上显示。如本说明书和本申请的任何权利要求项使用的是,术语“计算机可读介质(computer readable medium)”和“计算机可读介质(computer readable media)”完全局限于以通过计算机可读的形式存储信息的有形的、物理的对象。这些术语排除了任何无线信号、有线下载信号、和任何其它短暂信号。

[0066] 为了提供与用户的交互,能够在计算机上实现本说明书中描述的主题的实现方式,该计算机具有用于向用户显示信息的显示装置(例如,阴极射线管(CRT)或者液晶显示器(LCD)监视器);以及用户能够通过其向计算机提供输入的键盘和指向装置(例如,鼠标或者轨迹球)。其它种类的装置还能够被用于为与用户的交互;例如,提供给用户的反馈能够是任何例如视觉反馈、听觉反馈、或者触觉反馈的形式的感觉性反馈;以及能够以包括声音输入、语言输入、或者触觉输入的任何形式接收来自用户的输入而提供。此外,计算机能够

通过将文档发送至由用户使用的装置并且从由用户使用的装置接收文档(例如,通过响应于从web浏览器接收到的请求而将web页面发送到用户的客户端装置上的web浏览器)来与用户交互。

[0067] 本说明书中描述的主题的实施例能够在包括后端组件(例如,作为数据服务器)或包括中间件组件(例如,应用服务器)或包括前端组件(例如,具有图形用户界面或Web浏览器,用户能够通过其与本说明书中描述的主题的实现方式交互的用户计算机),或包括这样的后端组件、中间件组件、前端组件的一个或多个的任何组合的计算系统中实现。系统的组件能够通过数字数据通信的任何形式或介质(例如,通信网络)互连。通信网络的示例包括局域网(“LAN”)和广域网(“WAN”)、网络间网络(例如,互联网)、和对等网络(例如,自组织(ad hoc)对等网络)。

[0068] 计算系统能够包括客户端和服务端。客户端和服务端通常相互远离,并且典型地通过通信网络进行交互。客户端和服务端的关系凭借在相应计算机上运行并相互具有客户端-服务端关系的计算机程序而产生。在所公开的主题的一些方面中,服务端向用户装置传输数据(例如,HTML页面)(例如,出于向与客户端装置交互的用户显示数据并从与客户端装置交互的用户接收用户输入的目的)。能够在服务端处从客户端装置接收在客户端装置处生成的数据(例如,用户交互的结果)。

[0069] 理解的是,公开的过程中的步骤的任何特定的顺序或者层级是对示例方法的说明。基于设计偏好,理解的是,可以重新布置在过程中的特定步骤顺序或者层级,或者执行所有图示的步骤。可以同时执行这些步骤中的一些。例如,在某些情况下,多任务处理和并行处理可以是有利的。此外,不应当将以上图示的各种系统部件的分离理解为对这种分离的需要,并且应当理解的是,所描述的程序部件和系统通常能够一起被集成在单个软件产品中或者被封装到多个软件产品中。

[0070] 对这些方面的各种修改将会是显而易见的,并且本文中定义的通用原理可以被应用于其它方面。因此,权利要求不旨在限制于本文中示出的方面,而是应当赋予与语言权利要求一致的全部范围,其中,除非另有规定,否则以单数形式提及的元件不意味着“一个且仅一个”,而是指“一个或多个”。除非另有规定,术语“一些”是指一个或多个。男性代词(例如,他的)包括女性和中性性别(例如,她的和它的),反之亦然。标题和副标题(若存在)仅仅出于方便而使用并且不限制本主题技术。

[0071] 例如“方面”的短语并不意味着该方面对本主题技术是必要的或者并不意味着该方面应用于本主题技术的所有配置。与方面相关的公开可以应用于所有配置,或者一个或多个配置。例如方面的短语可以是指一个或多个方面,反之亦然。例如“配置”短语并不意味着这种配置对本主题技术是必要的或者并不意味着这种配置应用于本主题技术的所有配置。与配置相关的公开可以应用于所有配置,或者一个或多个配置。例如配置的短语可以是指一个或多个配置,反之亦然。

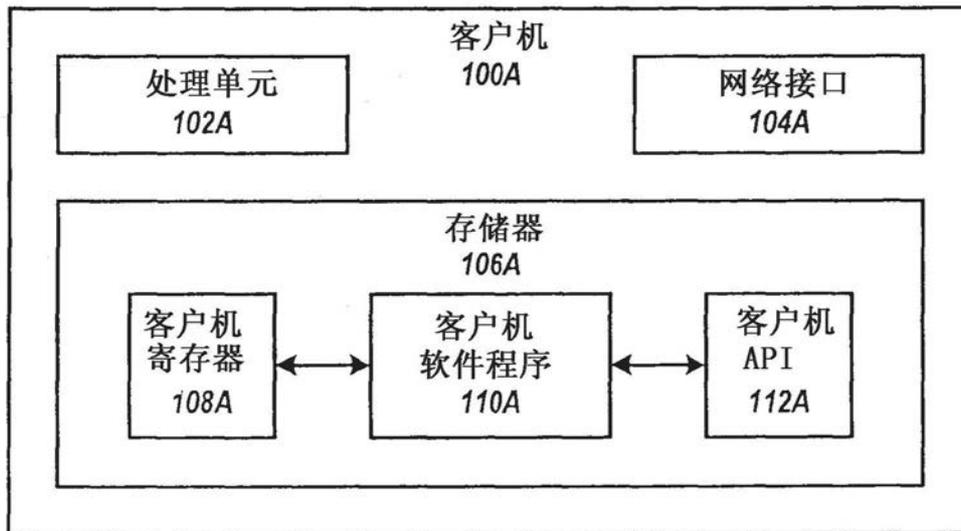


图1A

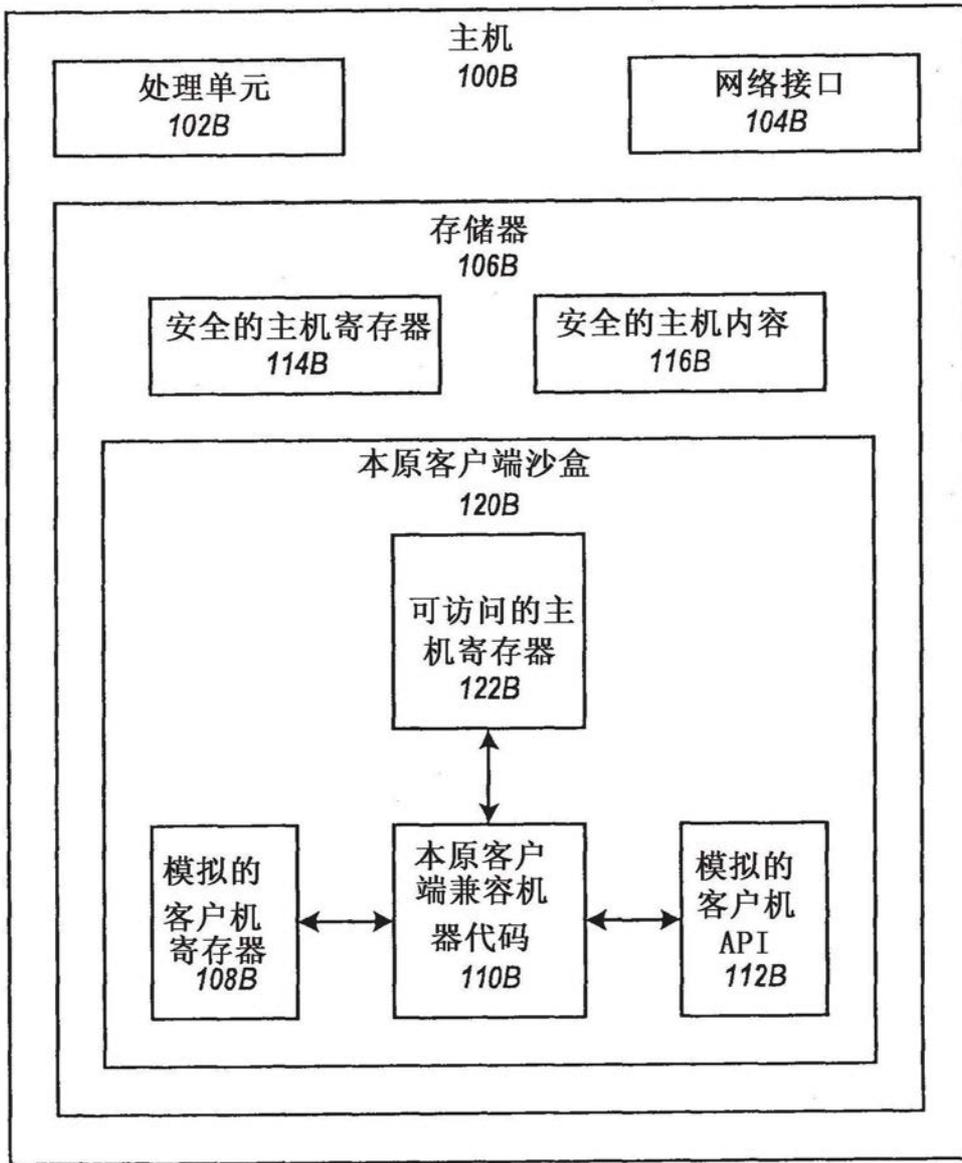


图1B

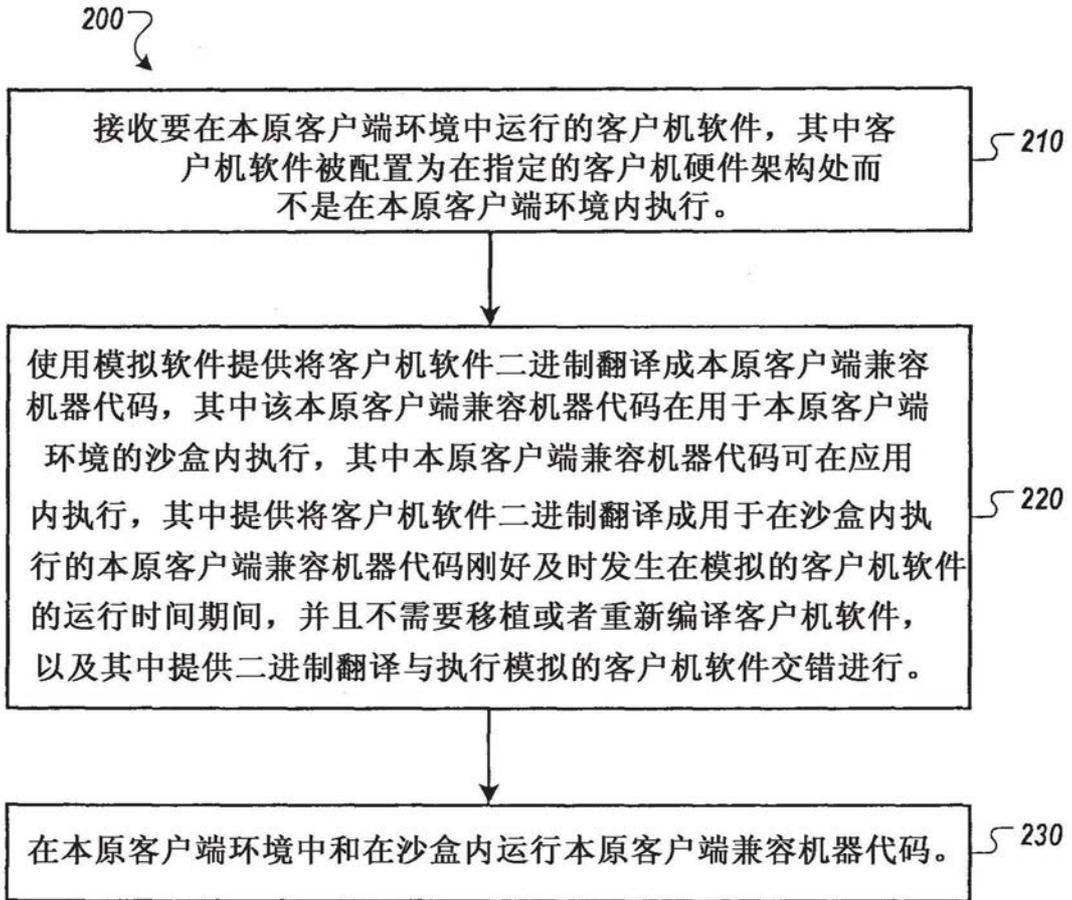


图2

300

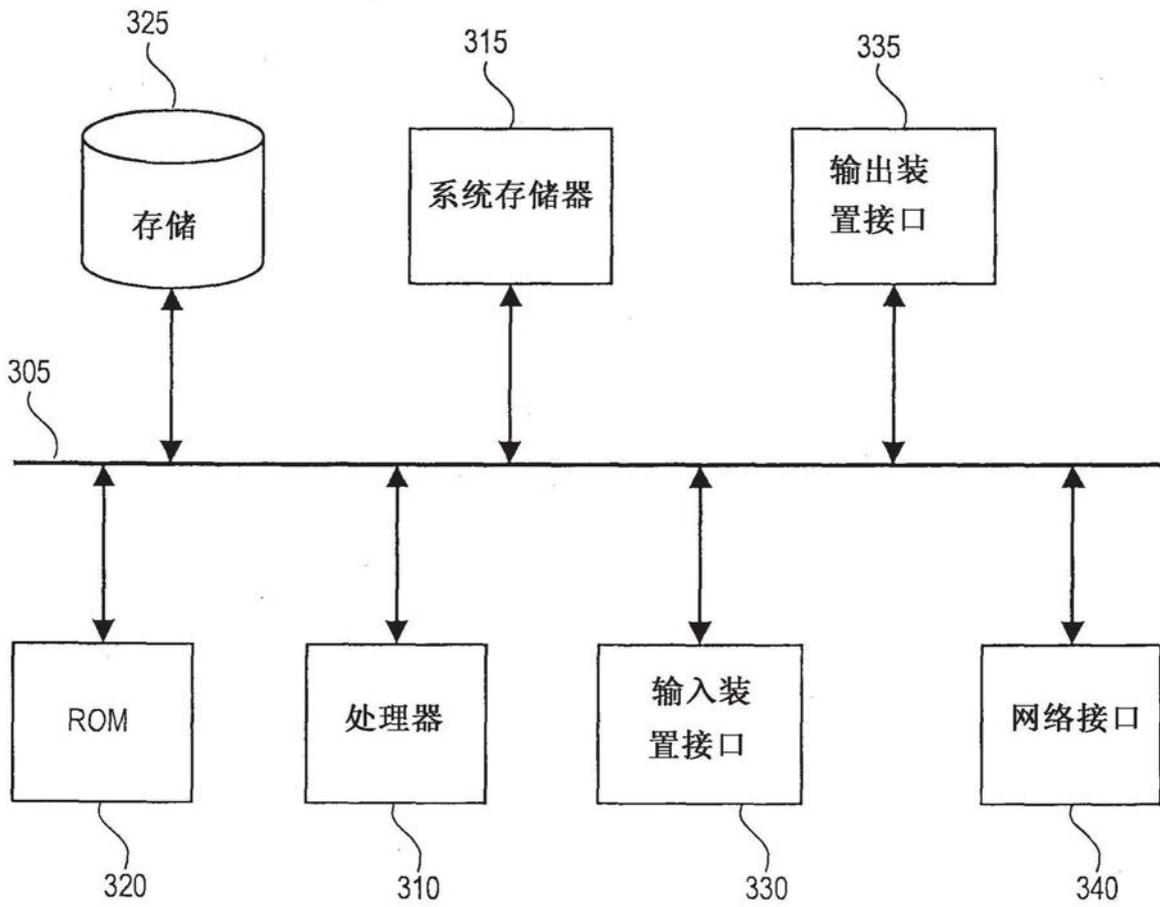


图3