



(12)发明专利申请

(10)申请公布号 CN 111831402 A

(43)申请公布日 2020. 10. 27

(21)申请号 201910300611.0

(22)申请日 2019.04.15

(71)申请人 戴尔产品有限公司

地址 美国得克萨斯州

(72)发明人 詹珺 唐文字 黄翔 刘伟 高寒
季乐威

(74)专利代理机构 北京市金杜律师事务所
11256

代理人 王茂华 罗利娜

(51) Int. Cl.

G06F 9/46(2006.01)

G06F 9/48(2006.01)

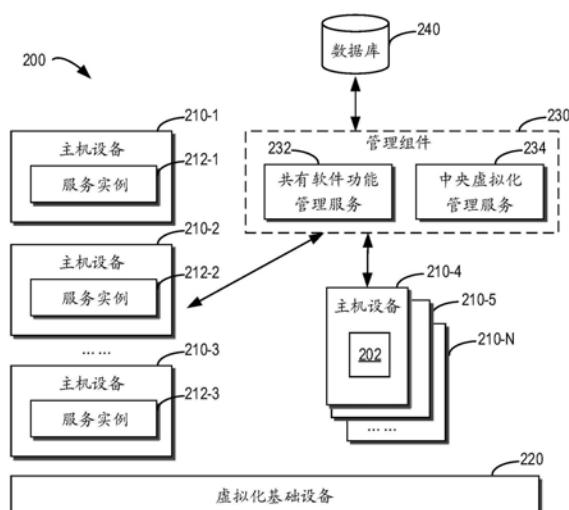
权利要求书3页 说明书11页 附图4页

(54)发明名称

用于管理软件功能的方法、设备和计算机程序产品

(57)摘要

本公开的实施例提供了用于管理软件功能的方法、设备和计算机程序产品。一种管理软件功能的方法包括：响应于确定服务实例要调用软件功能，向管理组件发起针对软件功能的调用请求，调用请求包括软件功能的标识信息和操作参数，软件功能由被部署在聚合基础架构内的服务实例和至少另一个服务实例共享并且独立于服务实例和至少另一个服务实例被存储在聚合基础架构中；以及从管理组件接收针对调用请求的响应，响应包括软件功能基于操作参数执行的结果。以此方式，无需随多个服务实例一起部署共有软件功能的多个副本，降低了性能和存储空间的耗费，并且还使软件功能的维护、更新和调用更方便。



1. 一种管理软件功能的方法,包括:

响应于确定服务实例要调用软件功能,向管理组件发起针对所述软件功能的调用请求,所述调用请求包括所述软件功能的标识信息和操作参数,所述软件功能由被部署在聚合基础架构内的所述服务实例和至少另一个服务实例共享并且独立于所述服务实例和所述至少另一个服务实例被存储在聚合基础架构中;以及

从所述管理组件接收针对所述调用请求的响应,所述响应包括所述软件功能基于所述操作参数执行的结果。

2. 根据权利要求1所述的方法,其中所述标识信息包括以下至少一项:所述软件功能的唯一标识符和所述软件功能在所述聚合基础架构内的访问路径。

3. 根据权利要求2所述的方法,其中所述标识信息还包括以下至少一项:所述软件功能的类型和存储所述软件功能的存储系统的名称。

4. 根据权利要求1所述的方法,其中发起所述调用请求包括:

从所述服务实例的属性文件加载所述标识信息和所述操作参数;以及

通过将所述标识信息和所述操作参数包括到所述调用请求中来发起所述调用请求。

5. 根据权利要求1所述的方法,其中所述融合基础架构包括超聚合基础架构。

6. 一种管理软件功能的方法,包括:

从请求方接收针对软件功能的调用请求,所述调用请求包括所述软件功能的标识信息和操作参数,所述软件功能由被部署在聚合基础架构内的多个服务实例共享并且独立于所述多个服务实例被存储;

响应于所述调用请求,使得由所述标识信息标识的所述软件功能基于所述操作参数而被执行;以及

向所述请求方发送针对所述调用请求的响应,所述响应包括所述软件功能基于所述操作参数执行的结果。

7. 根据权利要求6所述的方法,其中使得所述软件功能被执行包括:

验证所述请求方是否具有调用所述软件功能的权限;以及

响应于所述请求方具有所述权限,使得所述软件功能被执行。

8. 根据权利要求6所述的方法,其中使得所述软件功能被执行包括:

确定所述调用请求中包括的所述标识信息和所述操作参数中的至少一项是否符合所述软件功能的要求;以及

响应于确定所述标识信息和所述操作参数中的至少一项符合所述要求,使得所述软件功能被执行。

9. 根据权利要求6所述的方法,其中使得所述软件功能被执行包括:

基于所述标识信息来确定所述软件功能在聚合基础架构内的存储位置;以及

使得所述软件功能在所述存储位置处被执行。

10. 根据权利要求6所述的方法,其中所述标识信息包括以下至少一项:所述软件功能的标识符和所述软件功能的访问路径。

11. 根据权利要求10所述的方法,其中所述标识信息还包括以下至少一项:所述软件功能的类型和存储所述软件功能的存储系统的名称。

12. 一种电子设备,包括:

处理器;以及

与所述处理器耦合的存储器,所述存储器具有存储于其中的指令,所述指令在被处理器执行时使所述设备执行动作,所述动作包括:

响应于确定服务实例要调用软件功能,向管理组件发起针对所述软件功能的调用请求,所述调用请求包括所述软件功能的标识信息和操作参数,所述软件功能由被部署在聚合基础架构内的所述服务实例和至少另一个服务实例共享并且独立于所述服务实例和所述至少另一个服务实例被存储在聚合基础架构中;以及

从所述管理组件接收针对所述调用请求的响应,所述响应包括所述软件功能基于所述操作参数执行的结果。

13. 根据权利要求12所述的设备,其中所述标识信息包括以下至少一项:所述软件功能的唯一标识符和所述软件功能在所述聚合基础架构内的访问路径。

14. 根据权利要求13所述的设备,其中所述标识信息还包括以下至少一项:所述软件功能的类型和存储所述软件功能的存储系统的名称。

15. 根据权利要求12所述的设备,其中发起所述调用请求包括:

从所述服务实例的属性文件加载所述标识信息和所述操作参数;以及

通过将所述标识信息和所述操作参数包括到所述调用请求中来发起所述调用请求。

16. 根据权利要求12所述的设备,其中所述融合基础架构包括超聚合基础架构。

17. 一种电子设备,包括:

处理器;以及

与所述处理器耦合的存储器,所述存储器具有存储于其中的指令,所述指令在被处理器执行时使所述设备执行动作,所述动作包括:

从请求方接收针对软件功能的调用请求,所述调用请求包括所述软件功能的标识信息和操作参数,所述软件功能由被部署在聚合基础架构内的多个服务实例共享并且独立于所述多个服务实例被存储;

响应于所述调用请求,使得由所述标识信息标识的所述软件功能基于所述操作参数而被执行;以及

针对所述调用请求的响应,所述响应包括所述软件功能基于所述操作参数执行的结果。

18. 根据权利要求17所述的设备,其中使得所述软件功能被执行包括:

验证所述请求方是否具有调用所述软件功能的权限;以及

响应于所述请求方具有所述权限,使得所述软件功能被执行。

19. 根据权利要求17所述的设备,其中使得所述软件功能被执行包括:

确定所述调用请求中包括的所述标识信息和所述操作参数中的至少一项是否符合所述软件功能的要求;以及

响应于确定所述标识信息和所述操作参数中的至少一项符合所述要求,使得所述软件功能被执行。

20. 根据权利要求17所述的设备,其中使得所述软件功能被执行包括:

基于所述标识信息来确定所述软件功能在聚合基础架构内的存储位置;以及

使得所述软件功能在所述存储位置处被执行。

21. 根据权利要求17所述的设备,其中所述标识信息包括以下至少一项:所述软件功能的标识符和所述软件功能的访问路径。

22. 根据权利要求21所述的设备,其中所述标识信息还包括以下至少一项:所述软件功能的类型和存储所述软件功能的存储系统的名称。

23. 一种计算机程序产品,所述计算机程序产品被有形地存储在计算机可读介质上并且包括计算机可执行指令,所述计算机可执行指令在被执行时使设备执行根据权利要求1至5中任一项所述的方法。

24. 一种计算机程序产品,所述计算机程序产品被有形地存储在计算机可读介质上并且包括计算机可执行指令,所述计算机可执行指令在被执行时使设备执行根据权利要求6至11中任一项所述的方法。

用于管理软件功能的方法、设备和计算机程序产品

技术领域

[0001] 本公开的实施例涉及计算机技术,并且更具体地,涉及用于管理软件功能的方法、电子设备和计算机程序产品。

背景技术

[0002] 融合基础架构或超融合基础架构(HCI)是一种虚拟计算平台,该平台将计算、存储、网络 and 软件资源等融合成一个软件定义的基础架构。这样的虚拟计算平台已经获得越来越广泛的应用。虚拟化技术在其中被用于管理多个计算设备和存储系统。尽管从硬件层面看,该平台由多个物理设备和/或存储盘等构成,但通过集中虚拟化,不同设备/盘上的资源作为连续空间使用,连续空间诸如可以是存储虚拟化、硬件抽象、资源池创建等。在操作中,不同服务实例被部署在融合基础架构中运行。服务实例的运行可能需要使用一些软件功能,以提供相应的服务。

发明内容

[0003] 本公开的实施例提供了一种改进的管理软件功能的方案。

[0004] 在本公开的第一方面,提供了一种管理软件功能的方法。该方法包括:响应于确定服务实例要调用软件功能,向管理组件发起针对软件功能的调用请求,调用请求包括软件功能的标识信息和操作参数,软件功能由被部署在聚合基础架构内的服务实例和至少另一个服务实例共享并且独立于服务实例和至少另一个服务实例被存储在聚合基础架构中;以及从管理组件接收针对调用请求的响应,响应包括软件功能基于操作参数执行的结果。

[0005] 在本公开的第二方面,提供了一种管理软件功能的方法。该方法包括:从请求方接收针对软件功能的调用请求,调用请求包括软件功能的标识信息和操作参数,软件功能由被部署在聚合基础架构内的多个服务实例共享并且独立于多个服务实例被存储;响应于调用请求,使得由标识信息标识的软件功能基于操作参数而被执行;以及向请求方发送针对调用请求的响应,响应包括软件功能基于操作参数执行的结果。

[0006] 在本公开的第三方面,提供了一种电子设备。该电子设备包括处理器以及与处理器耦合的存储器,存储器具有存储于其中的指令,指令在被处理器执行时使设备执行动作,动作包括:响应于确定服务实例要调用软件功能,向管理组件发起针对软件功能的调用请求,调用请求包括软件功能的标识信息和操作参数,软件功能由被部署在聚合基础架构内的服务实例和至少另一个服务实例共享并且独立于服务实例和至少另一个服务实例被存储在聚合基础架构中;以及从管理组件接收针对调用请求的响应,响应包括软件功能基于操作参数执行的结果。

[0007] 在本公开的第四方面,提供了一种电子设备。该电子设备包括处理器以及与处理器耦合的存储器,存储器具有存储于其中的指令,指令在被处理器执行时使设备执行动作,动作包括:从请求方接收针对软件功能的调用请求,调用请求包括软件功能的标识信息和操作参数,软件功能由被部署在聚合基础架构内的多个服务实例共享并且独立于多个服务

实例被存储;响应于调用请求,使得由标识信息标识的软件功能基于操作参数而被执行;以及针对调用请求的响应,响应包括软件功能基于操作参数执行的结果。

[0008] 在本公开的第五方面,提供了一种计算机程序产品,计算机程序产品被有形地存储在计算机可读介质上并且包括计算机可执行指令,计算机可执行指令在被执行时使设备执行第一方面的方法。

[0009] 在本公开的第六方面,提供了一种计算机程序产品,计算机程序产品被有形地存储在计算机可读介质上并且包括计算机可执行指令,计算机可执行指令在被执行时使设备执行第二方面的方法。

[0010] 提供发明内容部分是为了简化的形式来介绍对概念的选择,它们在下文的具体实施方式中将被进一步描述。发明内容部分无意标识本公开的关键特征或主要特征,也无意限制本公开的范围。

附图说明

[0011] 通过结合附图对本公开示例性实施例进行更详细的描述,本公开的上述以及其它目的、特征和优势将变得更加明显,其中,在本公开示例性实施例中,相同的参考标号通常代表相同部件。

[0012] 图1示出了在常规的聚合基础架构的示意框图;

[0013] 图2示出了根据本公开的一些实施例的聚合基础架构的示意框图;

[0014] 图3示出了根据本公开的一些实施例的在聚合基础架构中管理软件功能的过程的流程图;

[0015] 图4示出了根据本公开的一些实施例的在聚合基础架构中管理软件功能的过程的流程图;

[0016] 图5示出了根据本公开的一些实施例的在软件功能的请求方处实现的过程的流程图;

[0017] 图6示出了根据本公开的一些实施例的在管理组件处实现的过程的流程图;以及

[0018] 图7示出了可以用来实施本公开的实施例的示例设备的框图。

具体实施方式

[0019] 下面将参考附图中示出的若干示例实施例来描述本公开的原理。虽然附图中显示了本公开的优选实施例,但应当理解,描述这些实施例仅是为了使本领域技术人员能够更好地理解进而实现本公开,而并非以任何方式限制本公开的范围。

[0020] 在本文中使用的术语“包括”及其变形表示开放性包括,即“包括但不限于”。除非特别申明,术语“或”表示“和/或”。术语“基于”表示“至少部分地基于”。术语“一个示例实施例”和“一个实施例”表示“至少一个示例实施例”。术语“另一实施例”表示“至少一个另外的实施例”。术语“第一”、“第二”等等可以指代不同的或相同的对象。下文还可能包括其他明确的和隐含的定义。

[0021] 在本文中,融合基础架构指的是一种虚拟计算平台,该平台将计算、存储、网络 and 软件资源等融合成一个软件定义的基础架构。在融合基础架构中,计算、存储、网络 and 软件资源是通过预先配置和集成的硬件组合提供,以作为连续空间使用。聚合基础架构通常包

含三种软件组件：计算虚拟化、存储虚拟化和管理。虚拟化软件可以抽象化和池化底层资源，并将它们动态分配给在虚拟机或容器中运行的服务实例。客户可以从单个工具管理计算、存储和网络连接。融合基础架构的一种示例演进被称为超融合基础架构(HCI)。

[0022] 在本文中，在聚合基础架构中的“服务”指的是经由聚合基础架构可访问的功能、处理、操作或能力等。服务的示例包括虚拟计算环境、虚拟操作系统、虚拟机、网络、服务器、数据存储中心、软件产品或应用程序等。服务在聚合基础架构中的实际部署或启动被称为“服务实例”。

[0023] 图1示出了常规聚合基础架构100的示意框图。如图1所述，在聚合基础架构100中部署有不同服务实例112-1、112-2、……112-N。为便于描述，这些服务实例被统称为或分别称为服务实例112。这些服务实例可以被部署在不同主机设备110-1、110-2、……110-N中。为便于描述，这些主机设备被统称为或分别称为主机设备110。

[0024] 服务实例112在从客户初始配置起到日常运行、状态校验等生命周期中均运行在所部署的主机设备110上。聚合基础架构100可以被构建在虚拟化基础设备120上。在聚合基础架构100还包括中央虚拟化管理服务130，其负责管理在聚合基础架构100中的服务部署、任务调度、资源分配等管理操作。

[0025] 运行在不同主机设备上的不同服务实例可能依赖于某个共同的软件功能来实现各自的工作。在聚合基础架构100中，各个主机设备110可能是物理隔离的。如果被部署在某个主机设备110上的服务实例112要使用另一个主机设备上的软件功能，该服务实例需要导入整体虚拟化依赖关系，以便定位软件功能、确定如何与具有该软件功能的另一个主机设备通信以及获得软件功能的执行结果。这会导致严重问题，因为在每个服务实例上维护整体虚拟化依赖关系所需的工作量非常大。

[0026] 因此，在常规聚合基础架构100中，每个服务实例112均具有软件功能的一个副本。如图1所示，各个服务实例112-1、112-2、……112-N均需要软件功能102。然而，多个副本的部署导致运行和存储空间的浪费，并且维护多个副本也是困难的。软件功能的相对位置会随着盘扩展或者其他物理环境的改变而改变。如果软件功能发生改变，对该软件功能的更新也是复杂且耗费成本的。此外，在这种情况下，客户调用由服务实例使用的软件功能的过程也非常复杂。例如，如果客户想要调用Java程序的Jar文件包中的软件功能，例如某个python脚本，客户需要首先登录到相应的主机设备，解压缩Jar文件包，提取出python脚本，并且指示脚本执行。

[0027] 根据本公开的实施例，提出了一种在聚合基础架构中改进的管理软件功能的方案。根据该方案，由多个服务实例共享的软件功能独立于服务实例被存储在聚合基础架构中。在服务实例要调用软件功能时，向管理组件发起针对软件功能的调用请求。调用请求中包括软件功能的标识信息和操作参数。管理组件响应于调用请求而使得软件功能基于操作参数而被执行，并且向服务实例返回软件功能的运行的结果。以此方式，无需随多个服务实例一起部署共有软件功能的多个副本，降低了性能和存储空间的耗费，并且还使软件功能的维护、更新和调用更方便。

[0028] 下面将参考附图来详细描述本公开的示例实施例。

[0029] 首先参考图2，其示出了根据本公开的一些实施例的聚合基础架构200的示意框图。聚合基础架构200可以被搭建在虚拟化基础设备220上，包括各种可以提供计算、存储、

网络和软件资源等的物理设备、存储盘等。这些基础设备220通过虚拟化技术构成聚合基础架构200。

[0030] 聚合基础架构200中还包括多个主机设备210-1、210-2、210-3、210-4、210-5、……210-N等。为便于描述,这些主机设备被统称为或分别称为主机设备210。在图2中示出的一个或多个主机设备210可以被理解为互相独立的物理设备或虚拟机。应当理解,虽然图2示出了一定数目的主机设备,但这个数目仅仅是一个示例。在其他实施例中,聚合基础架构200可以具有更多、更少或者任何其他数目的主机设备。

[0031] 根据本公开的实施例,多个服务实例212-1、212-2…212-3等被部署在聚合基础架构200中。为便于描述,这些服务实例被统称为或分别称为服务实例212。这些服务实例212被配置为提供相同或不同的服务。服务实例212可以响应于客户请求而被部署或者由管理服务根据需要而部署。多个服务实例212可以被部署在相同或不同主机设备210上。在图2的示例中,多个服务实例212-1、212-2、212-3被部署在不同主机设备210上。然而,应当理解,在同一个主机设备上部署多个服务实例也是可行的。

[0032] 在聚合基础架构中部署的两个或更多服务实例212可能需要使用同一个软件功能202来执行各自的任務。软件功能202可以服务实例212内需要调用以便实现特定功能的任何软件组件。软件功能的示例可以包括各种类型的脚本、可执行文件、安装包、操作系统资源包、小程序、进程、动态链接库(DLL)等等。本公开的实施例在此方面不受限制。在聚合基础架构中,软件功能202有时也可以被称为资源。由多个服务实例212共享的软件功能202因而也可称为共有资源或共有软件功能。

[0033] 不同于随着服务实例而部署共有软件功能的多个副本,根据本公开的实施例,软件功能202独立于多个服务实例212被存储。例如,在图2的示例中,软件功能202被存储在主机设备210-4中。在此,“独立于”指的是软件功能202与服务实例212具有一定程度的隔离,并且软件功能202无需作为服务实例212的一部分而必须被共同部署到同一主机设备210中。

[0034] 应当理解,图2仅示出了软件功能202的存储的一个示例。在其他实施例中,软件功能202可能被存储在任何其他主机设备210中,甚至可以被存储在同样部署有服务实例212的主机设备210。在存储在相同主机设备的场景中,软件功能202的存储与服务实例212的部署不存在必然联系,而是响应于管理层的集中管理而被分别存储和部署。在一些实现中,软件功能202可以被分布式存储在由多个主机设备210组成的集群中。本公开的实施例的范围在此方面不受限制。

[0035] 在聚合基础架构200还包括中央虚拟化管理服务230,其负责管理在聚合基础架构200中的服务部署、任务调度、资源分配等管理操作。在本公开的实施例中,管理组件230负责管理软件功能202。由于服务实例212与软件功能202之间的独立或隔离,在需要调用软件功能202时,服务实例212向管理组件230发送针对软件功能202的调用请求。取决于软件功能和实际应用需要,调用请求可以由服务实例212由于运行需要而自动发起,可以由服务实例212相关的管理进程发起,由管理服务器发起等等。在本公开的一些实施例中,以请求方来指示发起调用请求的源。

[0036] 管理组件230被配置为确定是否能够响应于调用请求而执行软件功能202。管理组件230可能需要查询数据库240来确定相关信息。在一些实施例中,管理组件230可以包括共

有软件功能管理服务模块232和中央虚拟化管理服务模块234。共有软件功能管理服务模块232被配置为管理多个服务实例212共有的软件功能202。在一些实施例中,共有软件功能管理服务模块232可以被配置为管理聚合基础架构200中的仅单个软件功能202,或者还可以被配置为管理由多个服务实例共有的一个或多个其他软件功能。中央虚拟化管理服务234被配置为管理聚合基础架构200中的其他功能,诸如实现服务部署、任务调度、资源分配等。

[0037] 为了能够调用被独立存储的软件功能202,服务实例212需要能够向管理组件230指示软件功能202。在服务实例212中,软件功能202可以由标识信息和服务实例212期望的操作参数来表示。在发起调用请求时,调用请求中包含这样的标识信息和操作参数。

[0038] 操作参数指的是特定服务实例212能够调用软件功能202所需要传递的参数,包括参数类型、参数的具体取值/名称等。操作参数的提供可以允许针对特定服务实例212来执行软件功能202。标识信息使得管理组件230能够标识软件功能202。在一些实施例中,标识信息可以包括软件功能202的唯一标识符,例如可以是一个通用唯一标识符(UUID)。标识信息还可以附加地或备选地包括软件功能在聚合基础架构内的访问路径,例如标识信息被存储在哪个主机设备上。

[0039] 在一些实施例中,标识信息还可以附加地或备选地包括存储软件功能的存储系统的名称或类似的标识信息。存储系统,或者也可称为数据存储集群、数据中心等,可能由多个主机设备的集群来实现。存储系统的名称使管理组件230可能知道如何在哪个存储系统中定位软件功能202。在指示存储系统的名称的情况下,访问路径可以指示在该存储系统中软件功能202的具体位置。在一些实施例中,标识信息还可以附加地或备选地包括软件功能202的名称。

[0040] 软件功能202的标识信息和操作参数可以被存储在服务实例212的属性文件中,该属性文件可以随着服务实例212的部署而被存储到相应的主机设备210中,或者可以被存储在任何一个可访问的存储位置。在一个实施例中,标识信息和操作参数可以以预定义的数据结构被组织在属性文件中和/或被包括在调用请求中。

[0041] 以下给出了包括软件202的标识信息和操作参数的一个示例数据结构。

[0042]

```
{  
  "resource_uuid": "7b11121c-0eb7-4000-9eb0-61c45a2  
36a28",  
  "data_store": "HGXGBD200000-01-J2-service-datasto  
re2",  
  "type": "python-script",  
  "path": "/pservice/payload/resource/diskscan/fetc  
h-disk-status.py",
```

```

    "name": " fetch-disk-status.py ",
    "allow-operation-parameter": {
        "disk-scan": {
            "parameters": [
                {
                    "type": "uuid",
                    "name": "disk-uuid"
                },
                {
                    "type": "string",
                    "name": "time-gap"
                }
            ]
        }
    }
}

```

[0043]

[0044] 在上述具体示例中,软件功能202是用于提取盘状态的功能,其形式是一个python脚本。软件功能202的标识信息包括UUID“7b11121c-0eb7-4000-9eb0-61c45a236a28”,存储软件功能202的存储系统的名称为“HGXGBD200000-01-J2-service-datastore2”,在存储系统内的访问路径是“/pservice/payload/resource/diskscan/fetch-disk-status.py”,软件功能202的类型为“python-script”(即python脚本)。软件功能202的操作参数指示类型为“uuid”的参数,该参数的名称为“disk-uuid”,指示要提取盘状态的盘标识。操作参数还可以包括类型为“string”(即字符串类型),该参数的名称为“time-gap”(即时间间隔),指示要提取盘状态的时间范围。

[0045] 应当理解,以上仅出于解释说明的目的而描述了软件功能的标识信息和操作参数的一个具体示例,不对本公开的范围有任何限制。在其他实施例中,取决于实际应用,其他标识信息和操作参数也是可能的。

[0046] 以上结合图2讨论了根据本公开的一些实施例的聚合基础架构。应当理解,图2仅是聚合基础架构的一个示意框图。聚合基础架构在逻辑上的示意框图还可以有其他表示形式和变型。在本公开的实施例的软件功能部署下,软件功能的请求方(诸如各个服务实例212)、软件功能的管理方(例如管理组件230)以及软件功能的执行方(例如,主机设备210-4)之间会存在交互。以下将参照图3和图4来继续描述交互过程。

[0047] 图3示出了根据本公开的一些实施例的在聚合基础架构中管理软件功能的过程300的流程图。过程300涉及服务实例212和管理组件230。为便于说明,将参考图2来描述过程300。

[0048] 在确定服务实例212要调用软件功能202时,从服务实例212向管理组件230发起310针对软件功能202的调用请求。如以上提及的,该调用请求包括软件功能的标识信息,诸如唯一标识符、类型、访问路径、存储系统的名称等一项或多项。这些标识信息不再赘述。调用请求还可以包括软件功能的操作参数。如以上提及的,除了图3示出的服务实例212本身之外,在备选实施例中,发起该调用请求的请求方还可以是管理进程、管理服务器或者甚至由客户手动发起。在一些实施例中,标识信息和操作参数可以从服务实例212的属性文件中

被加载出来并且被包括到调用请求中。

[0049] 响应于调用请求,管理组件230使得320由标识信息标识的软件功能202基于操作参数而被执行。在接收到调用请求后,管理组件230可以解析调用请求,以从中提取软件功能202的标识信息和操作参数。管理组件230通过标识信息定位而在聚合基础架构200中定位软件功能202,并且可以执行相应管理控制以使得软件功能202能够基于操作参数而被执行。例如,在图2的示例中,软件功能202被存储在主机设备210-4中,管理组件230可以在主机设备210-4上配置软件功能202所需的运行环境,包括软件和硬件资源等,并且在主机设备210-4上执行软件功能202。

[0050] 在一些实施例中,管理组件230中包括分离的共有软件功能管理服务模块232和中央虚拟化管理服务模块234,这两个模块之间具有一定交互以实现对软件功能202的调用的管理。在管理组件230内如何管理软件功能202的调用将在下文中参照图4详细讨论。

[0051] 在软件功能202的执行完成之后,管理组件230向请求方,在此是服务实例212,发送330针对调用请求的响应,响应包括软件功能202基于操作参数执行的结果。在一些实施例中,执行的结果的传送可以基于队列来实现。具体地,在发起调用请求之后,服务实例212发起队列进程,以监测来自管理组件230的消息队列。管理组件230在获得软件功能202的执行的结果后,可以将该结果放入服务实例212监测的队列中。由此,服务实例212可以通过队列进程的监测而获得执行的结果。基于队列的消息传送可以使用任何合适的消息队列协议,诸如高级消息队列协议 (AMQP) 或任何其他协议。基于队列的消息传送例如可以利用特定中间件,诸如Rabbit MQ或任何其他中间件来实现。除了基于队列的方案之外,管理组件230与服务实例212的消息交互还可以基于其他消息传送协议,诸如超文本传输协议 (HTTP) 或任何其他消息传送协议来实现。

[0052] 根据本公开的实施例,共用软件功能202的每个服务实例212均可以通过过程300来获得共有的软件功能202的执行结果,而无需将软件功能202的副本单独部署在本地主机设备中。相较于需要维护相同软件功能的多个副本,仅维护单个软件功能的成本将显著降低。当要部署新的服务实例时,如果存在与其他服务实例共用的软件功能,只需要将该软件功能的标识信息和操作参数随着服务实例的部署进行存储。这极大地提高了聚合基础设施的总体效率和稳定性。此外,这样的软件功能管理方式更方便由任何(授权)第三方能够方便地调用软件功能,简化了调用过程,提高了软件功能的可用性和调用的效率。这样的软件功能的调用在一些示例中可以被用作功能应用程序接口 (API)。

[0053] 接下来将参考图4详细讨论管理组件230在接收到调用请求后如何实现软件功能202的执行。图4示出了根据本公开的一些实施例的在聚合基础架构中管理软件功能的过程400的示意图。过程400涉及管理组件230中的共有软件功能管理服务模块232和中央虚拟化管理服务模块234以及存储软件功能202的主机设备210-4。为便于说明,将参考图2和图3来描述过程400。

[0054] 在过程400的实施例中,请求方(例如服务实例212)可以向共有软件功能管理服务模块232发起针对软件功能202的调用请求。接收到调用请求后,共有软件功能管理服务模块232可以解析调用请求,以从中提取软件功能202的标识信息和操作参数。

[0055] 共有软件功能管理服务模块232确定410调用请求中包括的标识信息和操作参数中的至少一项是否符合软件功能202的要求。例如,共有软件功能管理服务模块232可以从

数据库240获取软件功能202的要求,包括软件功能202的正确标识信息和/或标准操作参数定义。共有软件功能管理服务模块232可以通过比较来确定调用请求中的标识信息是否准确,和/或确定调用请求中的操作参数是否定义了正确或符合要求的类型、参数值等。由此,可以过滤掉一部分无效或错误的调用请求,避免触发后续操作。可以根据需要配置仅校验标识信息中的一项或多项,仅校验操作参数中的一项或多项,或者校验两者。

[0056] 如果共有软件功能管理服务模块232确定要校验的标识信息和操作参数中的至少一项符合要求,将可以触发软件功能的执行。具体地,共有软件功能管理服务模块232可以向中央虚拟化管理服务模块234发送420待执行的任务。该任务至少包括软件功能202的标识信息和操作参数。在接收到任务后,中央虚拟化管理服务模块234基于标识信息来确定430软件功能202在聚合基础架构内的存储位置。例如,在图2的示例中,中央虚拟化管理服务模块234可以根据标识信息定位出软件功能202被存储在主机设备210-4中。

[0057] 中央虚拟化管理服务模块234然后使得440软件功能202在存储位置(即主机设备210-4)处被执行。具体地,中央虚拟化管理服务模块234可以在主机设备210-4上配置软件功能202所需的运行环境,包括软件和硬件资源等,并且在主机设备210-4上运行共有软件功能管理服务模块232所指示的任务,包括使得软件功能202基于操作参数进行执行。在任务完成后,中央虚拟化管理服务模块234可以将软件功能202的执行的结果发送450给共有软件功能管理服务模块232。

[0058] 在一些实施例中,执行的结果的传送在中央虚拟化管理服务模块234与共有软件功能管理服务模块232之间的传送可以基于队列来实现。具体地,在发送任务之后,共有软件功能管理服务模块232发起一个队列进程,以监测来自中央虚拟化管理服务模块234的消息队列。在任务执行完毕之后,中央虚拟化管理服务模块234将执行的结果放入共有软件功能管理服务模块232监测的队列中。由此,共有软件功能管理服务模块232可以通过队列进程的监测而获得执行的结果。如以上提及的,基于队列的消息传送可以使用任何合适的消息队列协议。当然,除了基于队列的方案之外,还可以基于其他消息传送协议,诸如HTTP或任何其他消息传送协议来实现。

[0059] 在接收到任务的结果之后,共有软件功能管理服务模块232可以将接收到的任务的结果作为针对调用请求的响应而发送给服务实例212。如以上提及的,共有软件功能管理服务模块232可以将结果放入服务实例正在监听的队列中。备选地,也可以利用其它消息传送协议来将消息直接传送给服务实例212。

[0060] 在一些实施例中,如果共有软件功能管理服务模块232在检查调用请求时确定调用请求中的标识信息和操作参数中的至少一项不符合软件功能202的要求,共有软件功能管理服务模块232可以直接发送针对调用请求的响应,该响应指示调用请求不符合要求,无法被执行。由此,可以避免在错误调用请求的情况下服务实例212长时间无法监听到响应。

[0061] 在一些实施例中,由于共有的软件功能202可以方便地由第三方调用,管理组件230还可以利用应用一定的安全机制来确保软件功能202的安全性。管理组件230可以验证发起调用请求的请求方是否具有调用软件功能202的权限,并且在确定请求方具有权限的情况下使得软件功能202被执行。权限的验证可以基于各种安全机制,安全机制的示例可以包括开放授权(OAuth)标准或者任何其他适当的标准。取决于安全性要求,可以采用不同的安全策略。

[0062] 在一些实施例中,请求方,诸如服务实例212,可以被预先授权针对软件功能202的某个令牌。在发起调用请求时,服务实例可以携带该令牌。管理组件230可以从调用请求中提取该令牌,并且通过验证该令牌的有效性来确定请求方是否具有调用软件功能202的权限。

[0063] 在一些实施例中,管理组件230可以包括单独的安全服务模块(图2中未示出)用于实现权限验证。在另外一些实施例中,权限验证也可以由共有软件功能管理服务模块232或者中央虚拟化管理服务模块234来实现。本公开的实施例的范围在此方面不受限制。

[0064] 应当理解,虽然以上在管理组件230中分离的共有软件功能管理服务模块232和中央虚拟化管理服务模块234的实施例中讨论管理组件230如何管理软件功能202的执行,但是关于模块232和模块234中的一个模块描述的一些或全部功能可以由另一模块来实现。例如,可以不设置单独的共有软件功能管理服务模块232,而是由中央虚拟化管理服务模块234执行软件功能202的管理。又例如,可以由中央虚拟化管理服务模块234检验调用请求中的标识信息和/或操作参数,而共有软件功能管理服务模块232可以作为调用请求的接收和转发角色。本公开的实施例的范围在此方面不受限制。

[0065] 图5示出了根据本公开的一些实施例的在软件功能的请求方处实现的方法500的流程图。过程500可以被实现在图2的服务实例212处,或者管理服务实例212的管理进程或管理服务器等发起调用请求的请求方。为便于说明,参考图2来描述过程500。

[0066] 在框510,服务实例212响应于确定服务实例要调用软件功能,向管理组件发起针对软件功能的调用请求,调用请求包括软件功能的标识信息和操作参数,软件功能由被部署在聚合基础架构内的服务实例和至少另一个服务实例共享并且独立于服务实例和至少另一个服务实例被存储在聚合基础架构中。在框520,服务实例212从管理组件接收针对调用请求的响应,响应包括软件功能基于操作参数执行的结果。

[0067] 在一些实施例中,标识信息包括以下至少一项:软件功能的唯一标识符和软件功能在聚合基础架构内的访问路径。

[0068] 在一些实施例中,标识信息还包括以下至少一项:软件功能的类型和存储软件功能的存储系统的名称。

[0069] 在一些实施例中,发起调用请求包括:从服务实例的属性文件加载标识信息和操作参数;以及通过将标识信息和操作参数包括到调用请求中来发起调用请求。

[0070] 在一些实施例中,融合基础架构包括超聚合基础架构。

[0071] 图6示出了根据本公开的一些实施例的在管理组件处实现的方法600的流程图。过程600可以由图2的管理组件230实现。为便于说明,参考图2来描述过程600。

[0072] 在框610,管理组件230从请求方接收针对软件功能的调用请求,调用请求包括软件功能的标识信息和操作参数,软件功能由被部署在聚合基础架构内的多个服务实例共享并且独立于多个服务实例被存储。在框620,管理组件230响应于调用请求,使得由标识信息标识的软件功能基于操作参数而被执行。在框630,管理组件230向请求方发送针对调用请求的响应,响应包括软件功能基于操作参数执行的结果。在一些实施例中,请求方包括服务实例212。

[0073] 在一些实施例中,使得软件功能被执行包括:验证请求方是否具有调用软件功能的权限;响应于请求方具有权限,使得软件功能被执行。

[0074] 在一些实施例中,使得软件功能被执行包括:确定调用请求中包括的标识信息和操作参数中的至少一项是否符合软件功能的要求;以及响应于确定标识信息和操作参数中的至少一项符合要求,使得软件功能被执行。

[0075] 在一些实施例中,使得软件功能被执行包括:基于标识信息来确定软件功能在聚合基础架构内的存储位置;以及使得软件功能在存储位置处被执行。

[0076] 在一些实施例中,标识信息包括以下至少一项:软件功能的标识符和软件功能的访问路径。

[0077] 在一些实施例中,标识信息还包括以下至少一项:软件功能的类型和存储软件功能的存储系统的名称。

[0078] 图7示出了可以用来实施本公开的实施例的示例设备700的示意性框图。设备700可以被实现为或者被包括于图2的主机设备210、运行管理组件230的设备。

[0079] 如图所示,设备700包括中央处理单元(CPU)701,其可以根据存储在只读存储器(ROM)702中的计算机程序指令或者从存储单元708加载到随机访问存储器(RAM)703中的计算机程序指令,来执行各种适当的动作和处理。在RAM 703中,还可存储设备700操作所需的各种程序和数据。CPU 701、ROM 702以及RAM 703通过总线704彼此相连。输入/输出(I/O)接口705也连接至总线704。

[0080] 设备700中的多个部件连接至I/O接口705,包括:输入单元706,例如键盘、鼠标等;输出单元707,例如各种类型的显示器、扬声器等;存储单元708,例如磁盘、光盘等;以及通信单元709,例如网卡、调制解调器、无线通信收发机等。通信单元709允许设备700通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据。

[0081] 处理单元701执行上文所描述的各个方法和处理,例如过程500或过程600。例如,在一些实施例中,过程500或过程600可被实现为计算机软件程序或计算机程序产品,其被有形地包含于机器可读介质,诸如非瞬态计算机可读介质,诸如存储单元708。在一些实施例中,计算机程序的部分或者全部可以经由ROM 702和/或通信单元709而被载入和/或安装到设备700上。当计算机程序加载到RAM 703并由CPU 701执行时,可以执行上文描述的过程500或过程600的一个或多个步骤。备选地,在其他实施例中,CPU 701可以通过其他任何适当的方式(例如,借助于固件)而被配置为执行过程500或过程600。

[0082] 本领域的技术人员应当理解,上述本公开的方法的各个步骤可以通过通用的计算装置来实现,它们可以集中在单个的计算装置上,或者分布在多个计算装置所组成的网络上,可选地,它们可以用计算装置可执行的程序代码来实现,从而可以将它们存储在存储装置中由计算装置来执行,或者将它们分别制作成各个集成电路模块,或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。这样,本公开不限制于任何特定的硬件和软件结合。

[0083] 应当理解,尽管在上文的详细描述中提及了设备的若干装置或子装置,但是这种划分仅仅是示例性而非强制性的。实际上,根据本公开的实施例,上文描述的两个或更多装置的特征和功能可以在一个装置中具体化。反之,上文描述的一个装置的特征和功能可以进一步划分为由多个装置来具体化。

[0084] 以上所述仅为本公开的可选实施例,并不用于限制本公开,对于本领域的技术人员来说,本公开可以有各种更改和变化。凡在本公开的精神和原则之内,所作的任何修改、

等效替换、改进等,均应包含在本公开的保护范围之内。

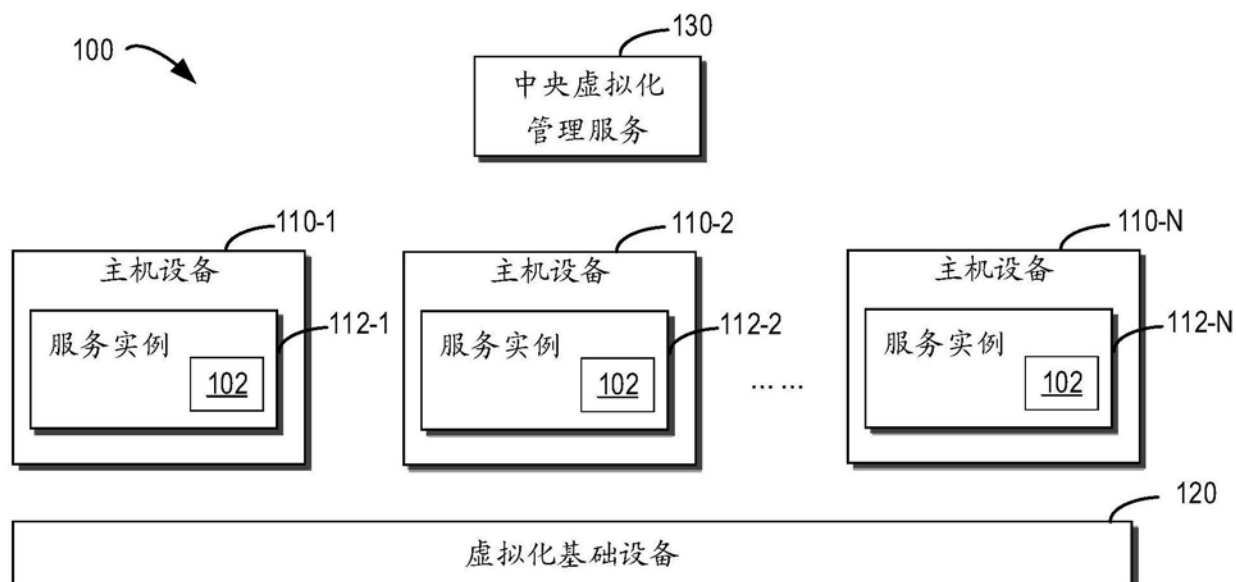


图1

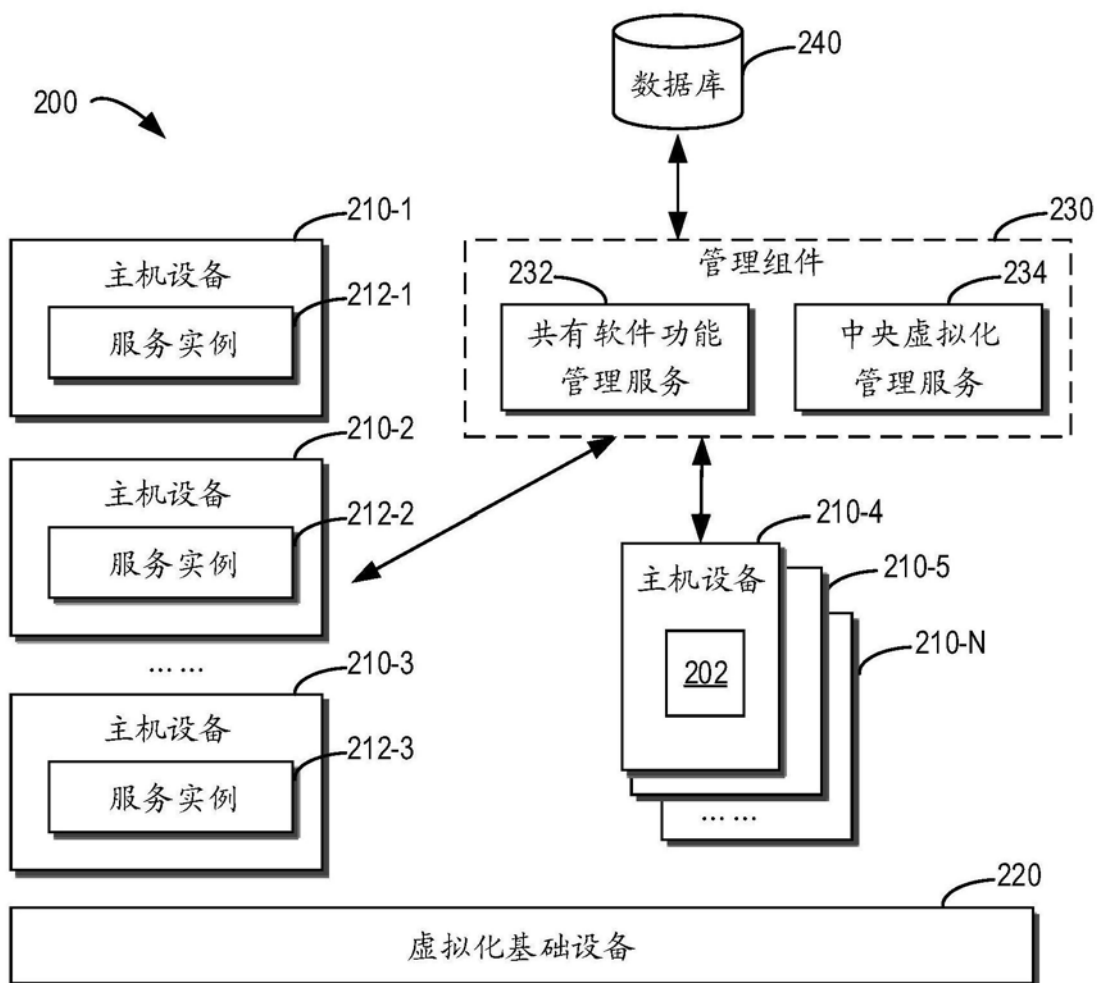


图2

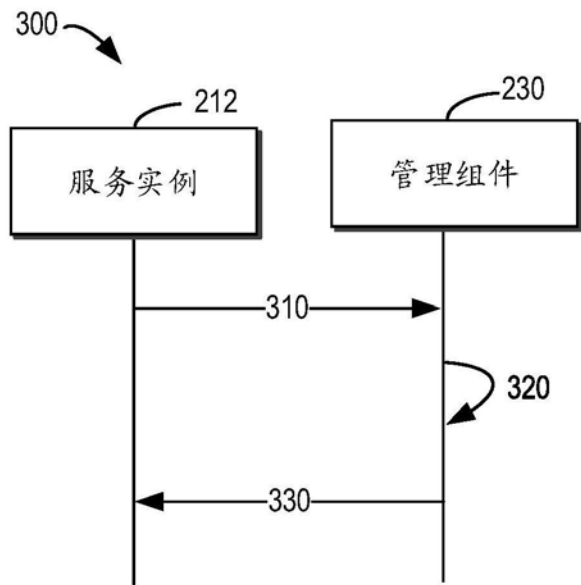


图3

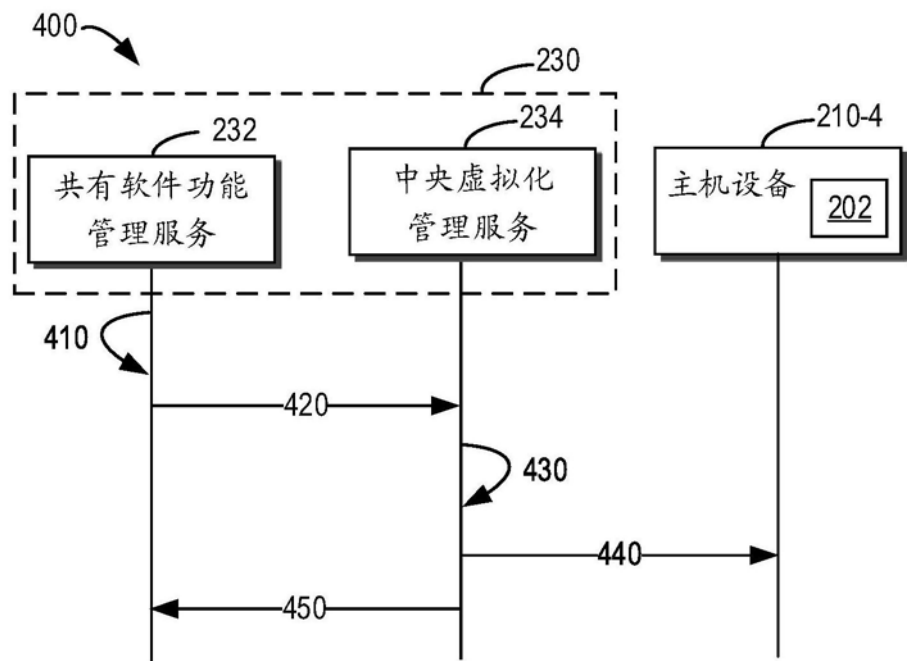


图4

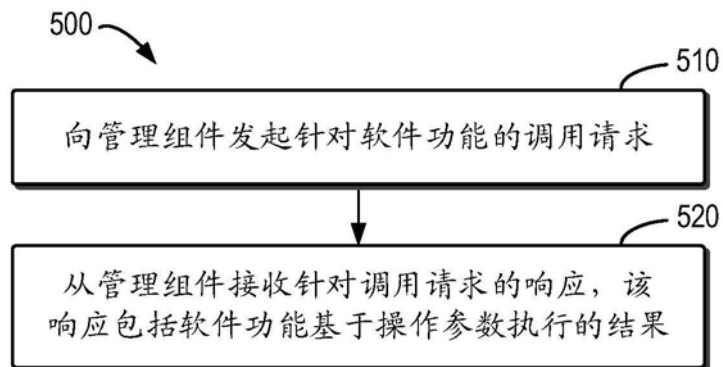


图5

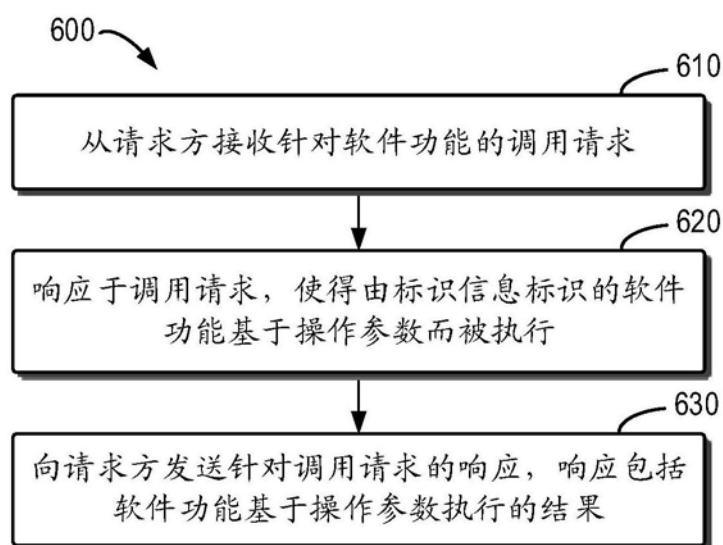


图6

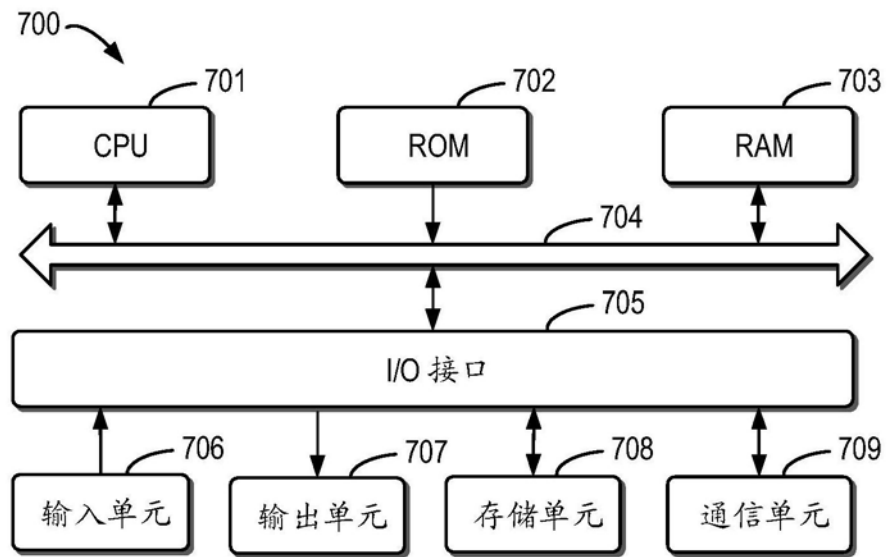


图7