



(12) 发明专利申请

(10) 申请公布号 CN 113934426 A

(43) 申请公布日 2022.01.14

(21) 申请号 202111115488.9

(22) 申请日 2021.09.23

(71) 申请人 北京车和家信息技术有限公司
地址 101300 北京市顺义区高丽营镇恒兴
路4号院1幢103室(科技创新功能区)

(72) 发明人 周鹏

(74) 专利代理机构 北京开阳星知识产权代理有
限公司 11710

代理人 吴崇

(51) Int.Cl.

G06F 8/41 (2018.01)

G06F 8/61 (2018.01)

G06F 11/36 (2006.01)

G06F 16/176 (2019.01)

G06F 21/64 (2013.01)

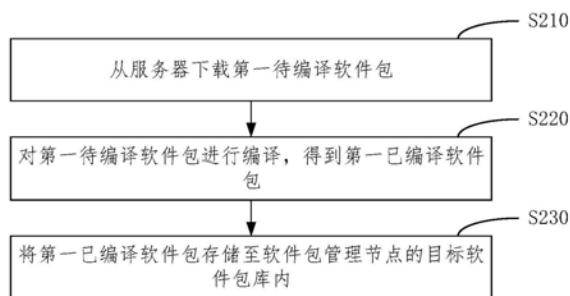
权利要求书3页 说明书17页 附图6页

(54) 发明名称

软件包处理方法、装置、系统、设备及介质

(57) 摘要

本公开涉及一种软件包处理方法、装置、系统、设备及介质。其中,软件包处理方法包括:从服务器下载第一待编译软件包;对第一待编译软件包进行编译,得到第一已编译软件包;将第一已编译软件包存储至软件包管理节点的目标软件包库内,目标软件包库为与目标应用平台相对应的软件包库,目标应用平台为对第一已编译软件包存在发布需求的应用平台。根据本公开实施例,能够在保证各个项目的软件包在独立存储的同时,通过软件包管理节点实现多个项目之间的软件包共享。



1. 一种软件包处理方法,其特征在于,包括:
 - 从服务器下载第一待编译软件包;
 - 对所述第一待编译软件包进行编译,得到第一已编译软件包;
 - 将所述第一已编译软件包存储至软件包管理节点的目标软件包库内,所述目标软件包库为与目标应用平台相对应的软件包库,所述目标应用平台为对所述第一已编译软件包存在发布需求的应用平台。
2. 根据权利要求1所述的方法,其特征在于,在所述对所述第一待编译软件包进行编译,得到第一已编译软件包之前,该方法还包括:
 - 分析所述第一待编译软件包所依赖的编译资源;
 - 若所述编译资源包括第二已编译软件包,则从所述软件包管理节点中提取所述第二已编译软件包;
 - 其中,所述对所述第一待编译软件包进行编译,得到第一已编译软件包,包括:
 - 基于所述第二已编译软件包,对所述第一待编译软件包进行编译,得到所述第一已编译软件包。
3. 根据权利要求1所述的方法,其特征在于,在所述将所述第一已编译软件包存储至软件包管理节点的目标软件包库内之后,该方法还包括:
 - 接收第一客户端发送的目标获取请求,所述目标获取请求携带有所述第一已编译软件包对应的软件包名称和所述第一已编译软件包对应的版本号;
 - 响应于所述目标获取请求,根据所述软件包名称和所述版本号,从所述软件包管理节点的所述目标软件包库中提取所述第一已编译软件包;
 - 将所述第一已编译软件包发送至所述第一客户端。
4. 根据权利要求1所述的方法,其特征在于,在所述将所述第一已编译软件包存储至软件包管理节点的目标软件包库内之后,该方法还包括:
 - 接收对所述目标应用平台的软件发布通知;
 - 响应于所述软件发布通知,从所述软件包管理节点的所述目标软件包库中提取最新版本的已编译软件包;
 - 将所述最新版本的已编译软件包发布至所述目标应用平台。
5. 根据权利要求1所述的方法,其特征在于,在所述将所述第一已编译软件包存储至软件包管理节点的目标软件包库内之前,所述方法还包括:
 - 为所述第一已编译软件包添加目标签名;
 - 其中,所述将所述第一已编译软件包存储至软件包管理节点的目标软件包库内,包括:
 - 将携带有所述目标签名的所述第一已编译软件包存储至所述软件包管理节点的所述目标软件包库内;
 - 其中,所述目标签名用于所述目标应用平台对所述第一已编译软件包进行签名校验,以使所述目标应用平台在签名校验通过后将所述第一已编译软件包发布至第二客户端。
6. 根据权利要求1所述的方法,其特征在于,在所述将所述第一已编译软件包存储至软件包管理节点的目标软件包库内之前,所述方法还包括:
 - 生成所述第一已编译软件包对应的校验信息;
 - 其中,所述将所述第一已编译软件包存储至软件包管理节点的目标软件包库内,包括:

将携带有所述校验信息的所述第一已编译软件包存储至所述软件包管理节点的所述目标软件包库内；

其中,所述校验信息用于所述应用平台目标应用平台对所述第一已编译软件包进行信息校验,以使所述应用平台目标应用平台在信息校验通过后将所述第一已编译软件包发布至第二客户端。

7.根据权利要求6所述的方法,其特征在于,所述校验信息包括软件包版本和信息摘要签名。

8.根据权利要求1所述的方法,其特征在于,在所述将所述第一已编译软件包存储至软件包管理节点的目标软件包库内之前,该方法还包括:

将所述第一已编译软件包发送至仿真节点进行仿真测试,得到所述第一已编译软件包对应的权限文件;

其中,所述将所述第一已编译软件包存储至软件包管理节点的目标软件包库内,包括:

将携带有所述权限文件的所述第一已编译软件包存储至所述软件包管理节点的所述目标软件包库内;

其中,所述权限文件包括运行所述第一已编译软件包所需授予的系统权限,所述权限文件用于第二客户端在运行所述第一已编译软件包时调用,第二客户端为接收到所述目标应用平台发布的所述第一已编译软件包的客户端。

9.根据权利要求1所述的方法,其特征在于,所述将所述第一已编译软件包存储至软件包管理节点的目标软件包库内,包括:

利用持续集成/持续部署技术,将所述第一已编译软件包存储至所述软件包管理节点的所述目标软件包库内。

10.一种软件包处理装置,其特征在于,包括:

软件包下载模块,配置为从服务器下载第一待编译软件包;

软件包编译模块,配置为对所述第一待编译软件包进行编译,得到第一已编译软件包;

软件包存储模块,配置为将所述第一已编译软件包存储至软件包管理节点的目标软件包库内,所述目标软件包库为与应用平台目标应用平台相对应的软件包库,所述应用平台目标应用平台为对所述第一已编译软件包存在发布需求的应用平台。

11.一种软件包处理系统,其特征在于,包括软件包开发节点和软件包管理节点;

所述软件包开发节点用于从服务器下载第一待编译软件包;对所述第一待编译软件包进行编译,得到第一已编译软件包;将所述第一已编译软件包存储至软件包管理节点的目标软件包库内,所述目标软件包库为与应用平台目标应用平台相对应的软件包库,所述应用平台目标应用平台为对所述第一已编译软件包存在发布需求的应用平台。

12.根据权利要求11所述的系统,其特征在于,所述系统还包括仿真节点;

所述软件包开发节点还用于将所述第一已编译软件包发送至仿真节点进行仿真测试,得到所述第一已编译软件包对应的权限文件;

所述软件包开发节点具体用于将携带有所述权限文件的所述第一已编译软件包存储至所述软件包管理节点的所述目标软件包库内;

其中,所述权限文件包括运行所述第一已编译软件包所需授予的系统权限,所述权限文件用于第二客户端在运行所述第一已编译软件包时调用,第二客户端为接收到所述目标

应用平台发布的所述第一已编译软件包的客户端。

13. 一种软件包处理设备,其特征在于,包括:

处理器;

存储器,用于存储可执行指令;

其中,所述处理器用于从所述存储器中读取所述可执行指令,并执行所述可执行指令以实现上述权利要求1-9中任一项所述的软件包处理方法。

14. 一种计算机可读存储介质,其特征在于,所述存储介质存储有计算机程序,当所述计算机程序被处理器执行时,使得处理器实现用上述权利要求1-9中任一项所述的软件包处理方法。

软件包处理方法、装置、系统、设备及介质

技术领域

[0001] 本公开涉及计算机技术领域,尤其涉及一种软件包处理方法、装置、系统、设备及介质。

背景技术

[0002] 在软件开发过程中,往往需要将源代码形式的软件包编译为适应于各应用平台的目标代码形式的软件包。

[0003] 现阶段,每个服务器在编译了软件包之后,会将已编译软件存储于该服务器对应的数据库中,从而导致各三方库包无法在多个软件开发项目之间共享,影响软件开发的效率。

发明内容

[0004] 为了解决上述技术问题或者至少部分地解决上述技术问题,本公开提供了一种软件包处理方法、装置、系统、设备及介质。

[0005] 第一方面,本公开提供了一种软件包处理方法,包括:

[0006] 从服务器下载第一待编译软件包;

[0007] 对第一待编译软件包进行编译,得到第一已编译软件包;

[0008] 将第一已编译软件包存储至软件包管理节点的目标软件包库内,目标软件包库为与目标应用平台相对应的软件包库,目标应用平台为对第一已编译软件包存在发布需求的应用平台。

[0009] 第二方面,本公开提供了一种软件包处理装置,包括:

[0010] 软件包下载模块,配置为从服务器下载第一待编译软件包;

[0011] 软件包编译模块,配置为对第一待编译软件包进行编译,得到第一已编译软件包;

[0012] 软件包存储模块,配置为将第一已编译软件包存储至软件包管理节点的目标软件包库内,目标软件包库为与应用平台目标应用平台相对应的软件包库,应用平台目标应用平台为对第一已编译软件包存在发布需求的应用平台。

[0013] 第三方面,本公开提供了一种软件包处理系统,包括:

[0014] 软件包开发节点和软件包管理节点;

[0015] 软件包开发节点用于从服务器下载第一待编译软件包;对第一待编译软件包进行编译,得到第一已编译软件包;将第一已编译软件包存储至软件包管理节点的目标软件包库内,目标软件包库为与应用平台目标应用平台相对应的软件包库,应用平台目标应用平台为对第一已编译软件包存在发布需求的应用平台。

[0016] 第四方面,本公开提供了一种软件包处理设备,包括:

[0017] 处理器;

[0018] 存储器,用于存储可执行指令;

[0019] 其中,处理器用于从存储器中读取可执行指令,并执行可执行指令以实现第一方

面提供的软件包处理方法。

[0020] 第五方面,本公开提供了一种计算机可读存储介质,该存储介质存储有计算机程序,当计算机程序被处理器执行时,使得处理器实现第一方面提供的软件包处理方法。

[0021] 本公开实施例提供的技术方案与现有技术相比具有如下优点:

[0022] 本公开实施例的软件包处理方法、装置、系统、设备及介质,可以在对服务器发送的第一待编译软件包进行编译之后,将编译得到的第一已编译软件包发送至软件包管理节点中与目标应用平台对应的目标软件包库,使得软件开发过程中对应于各应用平台的已编译软件包能够统一存储至软件包管理节点中,可以在保证各个项目的软件包在独立存储的同时,通过软件包管理节点实现多个项目之间的软件包共享。

附图说明

[0023] 结合附图并参考以下具体实施方式,本公开各实施例的上述和其他特征、优点及方面将变得更加明显。贯穿附图中,相同或相似的附图标记表示相同或相似的元素。应当理解附图是示意性的,原件和元素不一定按照比例绘制。

[0024] 图1示出了本公开实施例提供的一种软件包开发系统的系统框架图;

[0025] 图2示出了本公开实施例提供的一种软件包处理方法的流程示意图;

[0026] 图3示出了本公开实施例提供的一种示例性地软件包管理节点;

[0027] 图4是本公开实施例提供的另一种软件包处理方法的流程示意图;

[0028] 图5是本公开实施例提供的又一种软件包处理方法的流程示意图;

[0029] 图6是本公开实施例提供的一种示例性地提取第一已编译软件包的示意图;

[0030] 图7是本公开实施例提供的再一种软件包处理方法的流程示意图;

[0031] 图8是本公开实施例提供的再一种软件包处理方法的流程示意图;

[0032] 图9是本公开实施例提供的再一种软件包处理方法的流程示意图;

[0033] 图10是本公开实施例提供的再一种软件包处理方法的流程示意图;

[0034] 图11示出了本公开实施例提供的一种软件包处理装置的结构示意图;

[0035] 图12示出了本公开实施例提供的一种软件包处理系统的系统架构图;

[0036] 图13示出了本公开实施例提供的一种软件包处理设备的结构示意图。

具体实施方式

[0037] 下面将参照附图更详细地描述本公开的实施例。虽然附图中显示了本公开的某些实施例,然而应当理解的是,本公开可以通过各种形式来实现,而且不应该被解释为限于这里阐述的实施例,相反提供这些实施例是为了更加透彻和完整地理解本公开。应当理解的是,本公开的附图及实施例仅用于示例性作用,并非用于限制本公开的保护范围。

[0038] 应当理解,本公开的方法实施方式中记载的各个步骤可以按照不同的顺序执行,和/或并行执行。此外,方法实施方式可以包括附加的步骤和/或省略执行示出的步骤。本公开的范围在此方面不受限制。

[0039] 本文使用的术语“包括”及其变形是开放性包括,即“包括但不限于”。术语“基于”是“至少部分地基于”。术语“一个实施例”表示“至少一个实施例”;术语“另一实施例”表示“至少一个另外的实施例”;术语“一些实施例”表示“至少一些实施例”。其他术语的相关定

义将在下文描述中给出。

[0040] 需要注意,本公开中提及的“第一”、“第二”等概念仅用于对不同的装置、模块或单元进行区分,并非用于限定这些装置、模块或单元所执行的功能的顺序或者相互依存关系。

[0041] 需要注意,本公开中提及的“一个”、“多个”的修饰是示意性而非限制性的,本领域技术人员应当理解,除非在上下文另有明确指出,否则应该理解为“一个或多个”。

[0042] 本公开实施方式中的多个装置之间所交互的消息或者信息的名称仅用于说明性的目的,而并不是用于对这些消息或信息的范围进行限制。

[0043] 在软件的日常开发过程中,为了便于机器运行软件包的功能,软件开发人员在编写了源代码软件包之后,往往需要通过部署有开发环境的服务器上的交叉编译器,将源代码编译为各应用平台试用的目标代码软件包。在一个示例中,对于诸如车载嵌入式控制设备等嵌入式设备,由于嵌入式设备的性能局限,往往不能使用本机编译得到所需软件的可执行软件文件。因此,需要借助搭载有交叉编译器的服务器来对其进行编译。然而,每个服务器在编译了软件包之后,会将已编译软件存储于该服务器对应的数据库中,从而导致各服务器的三方库内的已编译软件包无法在多个软件开发项目之间共享,影响软件包的开发效率。

[0044] 基于此,本公开实施例提供了一种软件包处理方法、装置、系统、设备及介质,可以从服务器获取第一待编译软件包,以及对第一待编译软件包进行编译之后,将编译得到的第一已编译软件包发送至软件包管理节点中与目标应用平台对应的目标软件包库,使得软件开发过程中对应于各应用平台的已编译软件包能够统一存储至软件包管理节点中可以在保证各个项目的软件包在独立存储的同时,通过软件包管理节点实现多个项目之间的软件包共享。

[0045] 此外,在一些实施例中,交叉编译方式需要依赖大量的三方库。比如,当通过服务器A1编译软件包S1时,发现通过服务器A2编译的软件包S2是软件包S1的依赖资源,对于服务器S1而言,需要通过第三方库来调用软件包S2进行软件包S1的编译。这种编译方式往往因需要经过大量的处理、且可能存在着各种三方库无法共享等原因导致开发效率较低。

[0046] 而本公开实施例,在对软件包进行编译时,无需借助三方库即可从软件包管理节点中调用依赖资源,提高了软件包开发效率。

[0047] 首先,为了便于理解本公开实施例提供的方案,本公开实施例在开始介绍软件包处理方案之前,先对本公开实施例涉及的技术术语展开具体说明。

[0048] 开发环境,是指在基本硬件和数字软件的基础上,为支持系统软件和应用软件的工程化开发和维护而使用的一组软件。具体地,开发环境可以是用于软件的编译、调试、版本发布的环境,一般为软件包开发节点上运行的一组软件。

[0049] 软件包,是指具有特定的功能,用来完成特定任务的一个程序或一组程序。比如,可以是实现某个应用部分功能的程序代码的集合,或者,可以是实现应用平台的系统的部分功能的程序代码的集合。相应地,可以通过一个或者多个软件包,来实现对应用平台的系统或者应用进行更新或者安装。

[0050] 软件包编译,用于将源代码形式的软件包编译成目标代码形式的软件包。其中,源代码是指用汇编语言和/或高级语言写出来的、未经过编译的代码。目标代码是指源代码经过编译程序产生的能被应用平台识别的代码。

[0051] 在介绍了本公开实施例涉及的技术术语之后,为了便于从整体上了解本公开实施例提供的软件包处理方案,接下来,本公开实施例的下述部分先结合附图对本公开实施例涉及的软件包处理系统展开具体说明。

[0052] 图1示出了本公开实施例提供的一种软件包开发系统的系统框架图。

[0053] 如图1所示,软件包开发系统可以包括软件包处理系统10、服务器20以及应用平台30。其中,图1中的双向箭头所指向的双方可以进行通信或者数据交互。比如可以利用诸如有线网络、无线网络等进行交互,对此不作具体限定。

[0054] 接下来,本公开实施例的下述部分将依次对软件包开发系统的各组成设备或者系统展开具体说明。

[0055] 首先,对于软件包处理系统10,其可以对开发过程中的软件包进行开发以及管理。具体地,继续参见图1,软件包处理系统10包括软件包开发节点11和软件包管理节点12。

[0056] 其中,软件包开发节点11可以为软件包提供开发环境。在一些示例中,软件包开发节点11可以是物理服务器、虚拟服务器或者一段进程。在一个具体地示例中,软件包开发节点11可以是搭载有X64系统的虚拟机。需要说明的是,还可以根据实际场景和具体需求,使用搭载有诸如Linux系统或者Windows其他系统的虚拟机(比如X86等)作为软件包开发节点11,对此不作限定。

[0057] 在一些实施例中,软件包开发节点11所提供的开发环境能够提供诸如开发工具、依赖库、编译工具、仿真测试工具,来支持软件包的工程化开发和维护。

[0058] 其中,软件包管理节点12可以以应用平台为单位,对各应用平台对应的已编译软件包进行存储以及管理。在一些示例中,软件包管理节点12可以是物理服务器、虚拟服务器、数据库等,对软件包管理节点12的具体实现方式不作具体限定。

[0059] 其次,对于服务器20,可以是能够存储待编译软件包或者是获取软件开发人员在客户端上编写的待编译软件包的设备。在一些实施例中,服务器20可以是云服务器或者服务器集群等,对服务器20的具体实现方式不作具体限定。

[0060] 应用平台30,其用于为软件包提供展示、或者被下载通道的平台。在一些实施例中,应用平台30可以是硬件结构、浏览器、应用程序、软件框架、云计算平台、虚拟机、完整系统的虚拟化版本等。比如说应用市场、应用商店、下载中心、软件包发布平台等。在一些实施例中,多个第二客户端可以从同一应用平台下载软件包。从该应用平台获取的软件包能够在该多个第二客户端上执行。示例性地,具有相同操作系统的第二客户端可以通过该操作系统对应的应用平台下载相应的软件包。

[0061] 在一个具体的示例中,第二客户端可以是车辆内部具有控制功能、通信功能等的车载控制器。比如可以是车辆的中控台、整车控制器(Vehicle Control Unit,VCU)等。对于以电池为动力的电动车而言,第二客户端还可以是电池管理系统(Battery Management System,BMS)、电机控制器(Motor Control Unit,MCU)等车载控制器。可选地,上述车载控制器的系统可以是嵌入式系统。也就是说,车载控制器的系统或者应用软件与车载控制器的硬件一体化。

[0062] 在一个实施例中,一个软件包开发节点10可以与多个服务器20以及多个应用平台30进行交互。相应地,软件包开发服务器10可以为多个服务器20提供一个统一地、集成式的开发环境以及软件包管理节点。

[0063] 需要说明的是,本公开实施例提供的集成式的开发环境与现有技术相比至少具有如下优点中的至少一者:

[0064] 第一,集成式的开发环境可以提供每个开发人员开箱即用的开发环境,大大提升开发的效率。

[0065] 第二,由于无需在服务器上搭载开发环境,避免了因一般开发环境依赖特定的主机软件环境,需要执行繁琐的命令才能安装成功所导致的服务器开发环境搭建步骤比较繁琐的问题。

[0066] 第三,能够避免了开发人员为了在拉齐不同的目标平台的软硬件环境上花费大量的时间来组合和配置各种开发工具。

[0067] 第四,避免了多个服务器的开发环境难以完全一致所导致的不同服务器所提供的开发环境的接口偏差问题。

[0068] 第五,还可以对各应用平台的已编译软件包进行统一管理。

[0069] 第六,可以避免在各服务器上开发软件包且不同服务器无法实时共享第三方库的情况下,对各服务器造成的空间浪费。

[0070] 在一些实施例中,软件包处理系统12还可以包括仿真节点,仿真节点可以为软件包提供仿真测试环境。比如,仿真节点可以是物理服务器、虚拟服务器或者一段进程。在一个实施例中,为了实现对各应用平台的通用测试,仿真节点可以仿真与软件包开发系统交互的部分或者全部应用平台对应的客户端的测试环境。

[0071] 在一个实施例中,为了提高测试效率,仿真节点可以提供虚拟化的仿真测试环境。其中,虚拟化的仿真测试环境,可以是利用虚拟化技术在一定的软件以及硬件基础上构建的仿真测试环境。

[0072] 利用虚拟化的仿真测试环境进行软件包仿真的方案与现有技术中的利用真实的物理平台进行仿真的方案相比,物理平台往往又会依赖各种各样的外部硬件环境,搭建起来比较繁琐,测试过程比较耗时。而本公开实施例中利用虚拟化的仿真测试环境,可以在软件包开发完毕之后快速的在这个环境中进行验证功能,提升开发调试效率。

[0073] 在初步介绍了本公开实施例提供的软件包开发系统之后,本公开实施例接下来结合附图,依次对本公开实施例提供的软件包处理方法、装置、系统、设备及介质展开具体说明。

[0074] 图2示出了本公开实施例提供的一种软件包处理方法的流程示意图。其中,软件包处理方法各步骤的执行主体可以是软件包开发节点。其中,软件包开发节点的具体内容可以参见本公开上述实施例结合图1的相关说明,在此不再赘述。

[0075] 如图2所示,该软件包处理方法可以包括下述步骤S210至S230。

[0076] S210,从服务器下载第一待编译软件包。

[0077] 其中,对于第一待编译软件包,其可以是实现某个应用部分功能的源代码的集合,或者,可以是实现应用平台的系统的部分功能的源代码的集合。具体地,不同待编译软件包可以是不同系统或者不同应用功能的代码包,同一待编译软件包可以是同一应用功能或者程序的不同版本的软件包。在一些实施例中,服务器中的软件包可以是软件开发人员在客户端上完成待编译软件包的编写之后,将待编译软件包上传至服务器的。

[0078] 在一个示例中,第一待编译软件包可以是软件开发人员使用编程语言编写的、未

经编译的软件包。在一个具体的示例中,软件开发人员可以根据实际情况和具体需求,使用诸如C++、C、Java、Python等语言编写第一待编译软件包,本公开实施例对此不作具体限定。

[0079] 对于S210的具体实施方式,在一些实施例中,软件包开发节点可以每间隔一段时间从服务器下载新的、待编译软件包,并将所下载的新的、待编译软件包作为第一待编译软件包。其中,新的待编译软件包可以是上一次下载到本次下载之间产生的软件包。在另一些实施例中,服务器可以在确定存在新的待编译软件包之后,将其作为第一待编译软件包主动发送至软件包开发节点。本公开实施例对第一待编译软件包的下载方式不作具体限定。

[0080] S220,对第一待编译软件包进行编译,得到第一已编译软件包。

[0081] 对于S220,可以通过编译器,可以将第一待编译软件包的语言格式由源程序格式转换为能够被目标应用平台直接解读、运行的目标代码。在一些示例中,编译器可以是GCC、Clang、Visual Studio以及Eclipse等具有软件包编译功能的工具。

[0082] 在一个示例中,若目标应用平台为多个,则可以将第一待编译软件包编译为多个第一已编译软件包,使得每一个目标应用平台都有可在其上运行的第一已编译软件包。在一个具体的示例中,若待编译软件包S1的目标应用平台A、B、C的运行环境均不相同,则可以将待编译软件包S1编译为可以在目标应用平台A运行的已编译软件包S1_A、可以在目标应用平台B运行的已编译软件包S1_B、可以在目标应用平台C运行的已编译软件包S1_C。在另一个具体的示例中,为了提高编译效率,若目标应用平台A和目标应用平台B的运行环境相同,目标应用平台A与目标应用平台C的运行环境不同,则可以将待编译软件包S1编译为可以在目标应用平台A以及目标应用平台B运行的已编译软件包S1_{A/B}、可以在目标应用平台C运行的已编译软件包S1_C。

[0083] S230,将第一已编译软件包存储至软件包管理节点的目标软件包库内。

[0084] 其中,对于软件包管理节点,软件包管理节点内存在有与一个或者多个应用平台存在对应关系的软件包库,每个软件包库可以与至少一个应用平台相关联,例如每个软件包库的库名称可以携带有对应应用平台的标识信息。每一软件包库用于存储可以在与其对应的应用平台上运行的、已编译的软件包。在一个实施例中,每一应用平台都在软件包管理节点内存在与其对应的软件包库。可选地,为了提高资源利用率,不同运行环境的应用平台对应于不同的软件包库,相同运行环境的应用平台可以对应于同一软件包库。

[0085] 在一个示例中,图3示出了本公开实施例提供的一种示例性地软件包管理节点。如图3所示,软件包管理节点12可以包括多个软件包库,比如第一软件包库121、第二软件包库122以及第三软件包库123。

[0086] 其中,第一软件包库121对应于应用平台A,相应地,第一软件包库121内存储有对应于应用平台A的第一已编译软件包S1_A、第二已编译软件包S2_A、第三已编译软件包S3_A。

[0087] 第二软件包库122对应于应用平台B,相应地,第二软件包库122内存储有对应于应用平台B的第一已编译软件包S1_B、第二已编译软件包S2_B、第三已编译软件包S3_B。

[0088] 第三软件包库123对应于应用平台C,相应地,第三软件包库123内存储有对应于应用平台C的第一已编译软件包S1_C、第二已编译软件包S2_C、第三已编译软件包S3_C。

[0089] 在介绍了软件包管理节点以及软件包管理节点中的软件包库之后,接下来本公开的下述部分接着对目标软件包库以及目标应用平台展开具体说明。

[0090] 其中,目标应用平台为对第一已编译软件包存在发布需求的应用平台。也就是说,不同已编译软件可能对应于不同的目标应用平台。

[0091] 在一些实施例中,软件开发人员可以选择一个或者多个应用平台来发布第一待编译软件,其中,所选择的应用平台即为目标应用平台。在一个示例中,若软件开发人员在第一待编译软件的编写过程中选择了目标应用平台,则软件包开发平台还可以接收到目标应用平台的标识信息。示例性地,软件包管理节点可以查找库名称包含第一待编译软件包内携带的标识信息的软件包库,并将查找到的软件包库作为目标软件包库,进而将第一待编译软件包存储至目标软件包库内。

[0092] 在另一些实施例中,可以将预设的多个应用平台作为存在对第一已编译软件包存在发布需求的目标应用平台。比如,可以预先设置好应用平台A、应用平台B作为目标应用平台,当第一已编译软件包编译好之后,软件包管理节点可以查找库名称包含应用平台A的标识信息或者应用平台B的标识信息的软件包库,并将查找到的软件包库作为目标软件包库,进而将第一待编译软件包存储至目标软件包库内。

[0093] 在又一些实施例中,可以将存在第一待编译软件所对应的应用或者系统的应用平台作为目标应用平台。又比如,若编译得到第一待编译软件之后,若确定第一待编译软件可以被应用平台A和应用平台C对应的第二客户端运行,则可以将应用平台A和应用平台C作为目标应用平台,当第一已编译软件包编译好之后,软件包管理节点可以查找库名称包含应用平台A的标识信息或者应用平台C的标识信息的软件包库,并将查找到的软件包库作为目标软件包库,进而将第一待编译软件包存储至目标软件包库内。

[0094] 需要说明的是,本公开实施例还可以通过其他方式确定目标应用平台,对此不作具体限定。

[0095] 对于目标软件包库,目标软件包库为与目标应用平台相对应的软件包库。在一些实施例中,目标软件包库可以是软件包管理平台中的一个或者多个软件包库。示例性地,继续参见图3,若第二软件包库122和第三软件包库123对待编译软件S4存在发布需求,即第二软件包库122和第三软件包库123为待编译软件S4的目标应用平台。则可以将待编译软件S4对应于应用平台B的已编译软件包 $S4_b$ 存储至第二软件包库122,以及将对应于应用平台C的已编译软件包 $S4_c$ 存储至第三软件包库123。

[0096] 在一些实施例中,若目标应用平台在软件包管理平台不存在与其对应的软件包库,则软件包管理平台可以建立相应的软件包库。

[0097] 本公开实施例的软件包处理方法、装置、系统、设备及介质,可以在对服务器发送的第一待编译软件包进行编译之后,将编译得到的第一已编译软件包发送至软件包管理节点中与目标应用平台对应的目标软件包库,使得软件开发过程中对应于各应用平台的已编译软件包能够统一存储至软件包管理节点中,可以在保证各个项目的软件包在独立存储的同时,通过软件包管理节点实现多个项目之间的软件包共享。

[0098] 在一些实施例中,为了便于利用持续集成/持续部署技术,将第一已编译软件包存储至软件包管理节点的目标软件包库内。

[0099] 其中,对于持续集成,其表示可以频繁地、分多次将已通过测试的、已编译软件包存储至软件包管理节点。

[0100] 对于持续部署,其表示在已编译软件包通过测试后、自动部署至目标应用平台上。

在一些示例中,通过持续部署技术,可以将已编译软件自动存储至目标软件包库中,并从目标软件包库中获取已编译软件,将其自动部署至目标应用平台。

[0101] 通过本实施例,通过持续集成技术以及持续部署技术,已编译软件包必须通过自动化测试才能存储至软件包管理节点。从而使得软件包错误容易被快速发现以及定位,从而在软件包出问题之后无需消耗较多的时间去分析排查,提高了开发环境中软件包的迭代速度,缩短了软件包的迭代周期。

[0102] 此外,持续部署技术使得从开发人员提交代码到编译、测试、部署的全流程不需要人工的干预,完全通过自动化的方式进行软件开发。这一策略加快了已编译软件包存储至软件包管理节点的速度,保证新的已编译软件包能够第一时间部署到软件包管理节点。

[0103] 在一些可选地实施例中,基于上述持续集成技术/持续部署技术,可以在不删除旧版本的前提下,存储新的已编译软件包。在一个示例中,为了进一步便于对软件的溯源管理,在已编译软件包部署至软件包管理节点时,软件包管理节点可以生成该已编译软件包的目录,并将运行路径的符号链接指向该目标。当该版本的软件包出现问题时,可以通过修改符号链接,使其指向上一版本软件包的目录。从而通过本实施例,在软件包出问题,能够追溯响应软件包的版本和编译参数等内容。在一个示例中,可以基于Artifacts工具实现生成已编译软件包的目录。

[0104] 在本公开的一些实施例中,图4是本公开实施例提供的另一种软件包处理方法的流程示意图。

[0105] 如图4所示,该软件包处理方法可以包括下述步骤S410至S450。

[0106] S410,从服务器下载第一待编译软件包。其中,S410与图2中的S210相似,在此不做赘述。

[0107] S420,分析第一待编译软件包所依赖的编译资源。

[0108] 其中,编译资源可以是执行待编译任务时所需要的数据文件,例如编译资源可以包括图片、视频、文本、代码等,对编译资源的具体类型不作限定。其中,代码可以是除第一待编译软件包之外的其他软件包的代码或者是代码片段等。

[0109] S430,若编译资源包括第二已编译软件包,则从软件包管理节点中提取第二已编译软件包。

[0110] 在一些示例中,第二已编译软件包可以是在软件包管理阶段中存储在任意软件包库中的任意已编译软件包。示例性地,第二已编译软件包可以是第一待编译软件包的历史版本,或者是同一功能的其他软件包,或者是同一系统的其他软件包,或者是相关功能的软件包或者相关系统的软件包等,对其不作具体限定。

[0111] S440,基于第二已编译软件包,对第一待编译软件包进行编译,得到第一已编译软件包。其中,S440与图2中的S220相似,在此不做赘述。

[0112] S450,将第一已编译软件包存储至软件包管理节点的目标软件包库内。其中,S450与图2中的S230相似,在此不做赘述。

[0113] 通过本实施例,当对第一待编译软件包编译时,可以通过集成式的软件包管理节点提取第二已编译软件包,无需借助第三方库等,提高了软件开发效率。

[0114] 在本公开的一些实施例中,图5是本公开实施例提供的又一种软件包处理方法的流程示意图。

[0115] 如图5所示,该软件包处理方法可以包括下述步骤S510至S560。

[0116] S510,从服务器下载第一待编译软件包。其中,S510与图2中的S210相似,在此不做赘述。

[0117] S520,对第一待编译软件包进行编译,得到第一已编译软件包。其中,S520与图2中的S220相似,在此不做赘述。

[0118] S530,将第一已编译软件包存储至软件包管理节点的目标软件包库内。其中,S530与图2中的S230相似,在此不做赘述。

[0119] S540,接收第一客户端发送的目标获取请求。

[0120] 其中,目标获取请求用于请求获取一个或多个目标应用平台的第一已编译软件包。比如,若软件开发人员针对某一个或者多个目标应用平台进行应用开发时,或者是在软件开发人员需要查看某一个已编译软件包时通过第一客户端向软件包开发终端发送目标获取请求。

[0121] 具体地,目标获取请求可以携带有第一已编译软件包对应的软件包名称和第一已编译软件包对应的版本号。在一些示例中,为了获取某一个或者多个特定的目标应用平台的第一已编译软件包时,目标获取请求还可以携带有该一个或多个特定的目标应用平台的标识信息。

[0122] 其中,第一客户端可以是具有应用开关功能或者软件程序查看功能的设备,比如诸如平板电脑、智能手机、虚拟机、模拟器等具有人机交互功能的终端或者安装在终端上的提供人机交互入口的应用程序或者网页,对其具体类型不作限定。具体地,软件开发人员可以通过第一客户端向软件包开发节点请求已编译软件包。

[0123] S550,响应于目标获取请求,根据软件包名称和版本号,从软件包管理节点的目标软件包库中提取第一已编译软件包。其中,第一已编译软件包可以是第一客户端上存在开发需求的系统或者应用的一个或者多个软件包。

[0124] 在一个示例中,可以根据软件包名称和版本号,从软件包管理节点的目标软件包库中提取该版本号的第一已编译软件包。

[0125] 在另一个示例中,可以根据软件包名称、版本号以及一个或者多个特定的目标软件包库的标识信息,从一个或者多个特定的目标软件包库中提取该版本号的第一已编译软件包。

[0126] S560,将第一已编译软件包发送至第一客户端。

[0127] 在一个示例中,图6是本公开实施例提供的一种示例性地提取第一已编译软件包的示意图。如图6所示,当第一已编译软件包S1、第二已编译软件包S2、第三已编译软件包S3、第四已编译软件包S4通过持续集成技术或者持续部署技术部署至软件包管理平台12之后,第一客户端40需要对应用平台A开发时,可以从于应用平台A对应的目标应用平台121中提取第一已编译软件包S1_A。

[0128] 通过本实施例,可以在应用或者系统开发时,从软件包管理平台灵活提取相应地软件包,提高了应用或者系统的开发效率。

[0129] 在本公开的一些实施例中,图7是本公开实施例提供的再一种软件包处理方法的流程示意图。

[0130] 如图7所示,该软件包处理方法可以包括下述步骤S710至S760。

[0131] S710,从服务器下载第一待编译软件包。其中,S710与图2中的S210相似,在此不做赘述。

[0132] S720,对第一待编译软件包进行编译,得到第一已编译软件包。其中,S720与图2中的S220相似,在此不做赘述。

[0133] S730,将第一已编译软件包存储至软件包管理节点的目标软件包库内。其中,S730与图2中的S230相似,在此不做赘述。S740,接收对目标应用平台的软件发布通知。

[0134] 其中,软件发布通知可以是软件发布人员通过客户端发送至软件包开发平台的。具体地,软件发布通知用于通知软件包开发平台将目标应用平台的一个或者多个已编译软件包发布至目标应用平台。在一些实施例中,为了提高发布的精准度,软件发布通知可以包括目标应用平台的标识或者已编译软件包的名称。

[0135] 在一个示例中,可以在目标应用平台进行应用发布、应用部分功能升级、应用整体升级、系统部分功能升级、系统整体升级的场景中接收到针对目标应用平台的软件发布通知。

[0136] S750,响应于软件发布通知,从软件包管理节点的目标软件包库中提取最新版本的已编译软件包。

[0137] 在一些实施例中,可以根据目标应用平台的标识,从软件包管理节点查找到该目标应用平台对应的目标软件包库,然后根据已编译软件包的名称,从目标软件包库中查找到最新版本的已编译软件包。

[0138] S760,将最新版本的已编译软件包发布至目标应用平台。

[0139] 在一些实施例中,可以在目标软件包开发节点提供的部署环境中实现上述步骤S2740至S760。

[0140] 需要说明的是,为了实现开发环境和部署环境的有效隔离,可以用同一软件包开发节点的不同服务器或者不同进程来执行开发环境下的软件包处理步骤以及部署环境下的软件包处理步骤。

[0141] 因为部署环境往往需要一个目标平台下的所有应用、库、配置文件、分区等一起编译,时间往往会比较长。而不同部署平台的构建环境往往差异也是比较大的,比如可能会存在yocto、buildroot、ubuntu等,如果每一个应用都需要适配这些构建系统会导致应用的复杂度大大提升。通过本实施例,可以将开发环境跟部署环境进行有效的隔离,可以让开发环境更高效的迭代。

[0142] 在本公开的一些实施例中,图8是本公开实施例提供的再一种软件包处理方法的流程示意图。

[0143] 如图8所示,该软件包处理方法可以包括下述步骤S810至S840。

[0144] S810,从服务器下载第一待编译软件包。其中,S810与图2中的S210相似,在此不做赘述。

[0145] S820,对第一待编译软件包进行编译,得到第一已编译软件包。其中,S820与图2中的S220相似,在此不做赘述。

[0146] S830,为第一已编译软件包添加目标签名。

[0147] 在一些实施例中,为了保证开发过程的安全性,可以利用软件包开发平台中的安全组件对已编译好的软件包进行签名。具体地,安全组件可以利用设置于内部的私钥为第

一已编译软件包添加目标签名。

[0148] 示例性地,安全组件为加密服务器或者加密卡,安全组件还可以在对编译完成的软件包加目标签名之前,判断软件包是否为软件包开发平台编译的,如果是软件包开发平台编译的才会添加目标签名。可选地,为了提高安全性,安全组件中的私钥不可被读出。

[0149] 在一些实施例中,S830的具体实现方式可以包括:利用预设摘要算法,生成第一已编译软件包的摘要。然后利用加密服务器的私钥对摘要进行加密处理,得到目标签名。

[0150] S840,将携带有目标签名的第一已编译软件包存储至软件包管理节点的目标软件包库内。其中,S840与图2中的S830相似,在此不做赘述。

[0151] 其中,目标签名用于目标应用平台对第一已编译软件包进行签名校验,以使目标应用平台在签名校验通过后将第一已编译软件包发布至第二客户端。

[0152] 在一个实施例中,目标应用平台和安全组件处于同一公钥基础设施(Public Key Infrastructure,PKI)体系下。可以将安全组件的公钥分发至同一PKI体系下的应用平台。示例性地,可以在应用平台出厂的时候将安全组件的公钥预置到应用平台中。为了提高安全性,出厂之后的应用平台中的安全组件的公钥不可被更改且受到相应的安全保护。

[0153] 相应地,目标应用平台的签名校验过程可以具体包括:利用安全组件的公钥对目标签名进行解密,得到待验证摘要。利用与加密服务器相同的预设摘要算法计算第一已编译软件包的可靠摘要。在可靠摘要与待验证摘要一致的情况下,确定目标签名校验通过。

[0154] 通过本实施例,在目标应用平台的控制下,不会有恶意软件包下载到第二客户端上,避免了对第二客户端的伤害。

[0155] 在本公开的一些实施例中,图9是本公开实施例提供的再一种软件包处理方法的流程示意图。

[0156] 如图9所示,该软件包处理方法可以包括下述步骤S910至S940。

[0157] S910,从服务器下载第一待编译软件包。其中,S910与图2中的S210相似,在此不做赘述。

[0158] S920,对第一待编译软件包进行编译,得到第一已编译软件包。其中,S920与图2中的S220相似,在此不做赘述。

[0159] S930,生成第一已编译软件包对应的校验信息。

[0160] S940,将携带有校验信息的第一已编译软件包存储至软件包管理节点的目标软件包库内。

[0161] 其中,校验信息用于目标应用平台对第一已编译软件包进行信息校验,以使目标应用平台在信息校验通过后将第一已编译软件包发布至第二客户端。

[0162] 在一些实施例中,校验信息可以是能够体现已编译软件包的安全性、完整性、正确性、真实性等特性的信息。示例性地,校验信息可以包括信息软件包版本和信息摘要签名。其中,软件包版本用于校验第一已编译软件包是否是最新版本。信息摘要签名可以对第一已编译软件包的完整性进行校验。以及还可以通过对第一已编译软件包的完整性校验,第一已编译软件包是否被篡改过。

[0163] 通过本实施例,可以通过对第一已编译软件包对应的校验信息校验的方式,可以提高软件部署过程的可靠性。

[0164] 在本公开的一些实施例中,图10是本公开实施例提供的再一种软件包处理方法的

流程示意图。

[0165] 如图10所示,该软件包处理方法可以包括下述步骤S1010至S1040。

[0166] S1010,从服务器下载第一待编译软件包。其中,S1010与图2中的S210相似,在此不做赘述。

[0167] S1020,对第一待编译软件包进行编译,得到第一已编译软件包。其中,S1020与图2中的S220相似,在此不做赘述。

[0168] S1030,将第一已编译软件包发送至仿真节点进行仿真测试,得到第一已编译软件包对应的权限文件。

[0169] 其中,权限文件包括运行第一已编译软件包所需授予的系统权限,权限文件用于第二客户端在运行第一已编译软件包时调用,第二客户端为接收到目标应用平台发布的第一已编译软件包的客户端。

[0170] 在一些实施例中,权限文件可以包括:在客户端上运行第一已编译软件包时需要用户决定是否授予第一已编译软件包的权限。比如,读取通讯录、设备信息、麦克风、照片、定位等权限,对其不作具体限定。

[0171] 在一些实施例中,S1030可以具体实现为:通过自动或者手动地在仿真节点上运行第一已编译软件包,以在仿真运行的过程中,判断运行软件包需要系统的哪些权限,并且生成权限文件。

[0172] S1040,将携带有权限文件的第一已编译软件包存储至软件包管理节点的目标软件包库内。

[0173] 可选地,第二客户端在运行第一已编译软件包时调用时,运行软件包的操作系统可以向用户请求权限文件中所涉及的权限,如果用户拒绝授权,则无法运行该软件包。

[0174] 通过本实施例,通过仿真权限检查,可以在真正运行第一已编译软件包之前,确定其运行权限,提高了运行安全性。

[0175] 基于相同的发明构思,本公开实施例还提供了一种软件包处理装置,下面结合图11进行说明。

[0176] 图11示出了本公开实施例提供的一种软件包处理装置的结构示意图。在本公开的一些实施例中,该软件包处理装置可以实现为软件包开发节点。

[0177] 如图11所示,该种软件包处理装置1100,可以包括软件包下载模块1110、软件包编译模块1120和软件包存储模块1130。

[0178] 软件包下载模块1110,配置为从服务器下载第一待编译软件包;

[0179] 软件包编译模块1120,配置为对第一待编译软件包进行编译,得到第一已编译软件包;

[0180] 软件包存储模块1130,配置为将第一已编译软件包存储至软件包管理节点的目标软件包库内,目标软件包库为与应用平台目标应用平台相对应的软件包库,应用平台目标应用平台为对第一已编译软件包存在发布需求的应用平台。

[0181] 本公开实施例的软件包处理装置,可以在对服务器发送的第一待编译软件包进行编译之后,将编译得到的第一已编译软件包发送至软件包管理节点中与目标应用平台对应的目标软件包库,使得软件开发过程中对应于各应用平台的已编译软件包能够统一存储至软件包管理节点中,可以在保证各个项目的软件包在独立存储的同时,通过软件包管理节

点实现多个项目之间的软件包共享。

[0182] 在本公开一些实施例中,软件包处理装置1100还包括资源分析模块以及软件包提取模块。

[0183] 资源分析模块,配置为分析第一待编译软件包所依赖的编译资源;

[0184] 软件包提取模块,配置为若编译资源包括第二已编译软件包,则从软件包管理节点中提取第二已编译软件包;

[0185] 相应地,软件包编译模块1120,可以具体被配置为:

[0186] 基于第二已编译软件包,对第一待编译软件包进行编译,得到第一已编译软件包。

[0187] 在本公开一些实施例中,软件包处理装置1100还包括:请求接收模块、软件包提取模块和软件包发送模块。

[0188] 请求接收模块,配置为接收第一客户端发送的目标获取请求,目标获取请求携带有第一已编译软件包对应的软件包名称和第一已编译软件包对应的版本号;

[0189] 软件包提取模块,配置为响应于目标获取请求,根据软件包名称和版本号,从软件包管理节点的目标软件包库中提取第一已编译软件包;

[0190] 软件包发送模块,配置为将第一已编译软件包发送至第一客户端。

[0191] 在本公开一些实施例中,软件包处理装置1100还包括通知接收模块、软件包提取模块和软件包发布模块。

[0192] 通知接收模块,配置为接收对目标应用平台的软件发布通知;

[0193] 软件包提取模块,配置为响应于软件发布通知,从软件包管理节点的目标软件包库中提取最新版本的已编译软件包;

[0194] 软件包发布模块,配置为将最新版本的已编译软件包发布至目标应用平台。

[0195] 在本公开一些实施例中,软件包处理装置1100还包括:签名添加模块。

[0196] 签名添加模块,配置为为第一已编译软件包添加目标签名;

[0197] 相应地,软件包存储模块1130具体配置为:

[0198] 将携带有目标签名的第一已编译软件包存储至软件包管理节点的目标软件包库内;

[0199] 其中,目标签名用于目标应用平台对第一已编译软件包进行签名校验,以使目标应用平台在签名校验通过后将第一已编译软件包发布至第二客户端。

[0200] 在本公开一些实施例中,软件包处理装置1100还包括:校验信息生成模块。

[0201] 校验信息生成模块,配置为生成第一已编译软件包对应的校验信息;

[0202] 相应地,软件包存储模块1130具体配置为:

[0203] 将携带有校验信息的第一已编译软件包存储至软件包管理节点的目标软件包库内;

[0204] 其中,校验信息用于应用平台目标应用平台对第一已编译软件包进行信息校验,以使应用平台目标应用平台在信息校验通过后将第一已编译软件包发布至第二客户端。

[0205] 在本公开一些实施例中,校验信息包括软件包版本和信息摘要签名。

[0206] 在本公开一些实施例中,软件包处理装置1100还包括:权限文件获取模块。

[0207] 权限文件获取模块,配置为将第一已编译软件包发送至仿真节点进行仿真测试,得到第一已编译软件包对应的权限文件;

[0208] 相应地,软件包存储模块1130具体配置为:

[0209] 将携带有权限文件的第一已编译软件包存储至软件包管理节点的目标软件包库内;

[0210] 其中,权限文件包括运行第一已编译软件包所需授予的系统权限,权限文件用于第二客户端在运行第一已编译软件包时调用,第二客户端为接收到目标应用平台发布的第一已编译软件包的客户端。

[0211] 在本公开一些实施例中,软件包存储模块1130具体配置为:

[0212] 利用持续集成/持续部署技术,将第一已编译软件包存储至软件包管理节点的目标软件包库内。

[0213] 需要说明的是,图11所示的软件包处理装置1100可以执行图2至图10所示的方法实施例中的各个步骤,并且实现图2至图10所示的方法实施例中的各个过程和效果,在此不做赘述。

[0214] 基于相同的发明构思,本公开实施例还提供了一种软件包处理系统,下面结合图12进行说明。

[0215] 图12示出了本公开实施例提供的一种软件包处理系统的系统架构图。

[0216] 如图12所示,该软件包处理系统1200,可以包括软件包开发平台1210以及软件包管理平台1220。

[0217] 软件包开发节点1210用于从服务器下载第一待编译软件包;对第一待编译软件包进行编译,得到第一已编译软件包;将第一已编译软件包存储至软件包管理节点的目标软件包库内,目标软件包库为与应用平台目标应用平台相对应的软件包库,应用平台目标应用平台为对第一已编译软件包存在发布需求的应用平台。

[0218] 需要说明的是,软件包开发节点1210和软件包管理平台1220的其他内容可以参见本公开实施例上述部分结合图1至图11示出的软件包开发系统、软件包处理方法、软件包处理方法的相关内容,在此不再限定。

[0219] 本公开实施例的软件包处理系统,可以对服务器发送的第一待编译软件包进行编译之后,将编译得到的第一已编译软件包发送至软件包管理节点中与目标应用平台对应的目标软件包库,使得软件开发过程中对应于各应用平台的已编译软件包能够统一存储至软件包管理节点中,可以在保证各个项目的软件包在独立存储的同时,通过软件包管理节点实现多个项目之间的软件包共享。

[0220] 在本公开一些实施例中,软件包处理系统1200还包括仿真节点;

[0221] 软件包开发节点1210还用于将第一已编译软件包发送至仿真节点进行仿真测试,得到第一已编译软件包对应的权限文件;

[0222] 软件包开发节点1210具体用于将携带有权限文件的第一已编译软件包存储至软件包管理节点的目标软件包库内;

[0223] 其中,所述权限文件包括运行所述第一已编译软件包所需授予的系统权限,所述权限文件用于第二客户端在运行所述第一已编译软件包时调用,第二客户端为接收到所述目标应用平台发布的所述第一已编译软件包的客户端。

[0224] 需要说明的是,图12所示的软件包处理系统1200可以执行图2至图10所示的方法实施例中的各个步骤,并且实现图2至图10所示的方法实施例中的各个过程和效果,在此不

做赘述。

[0225] 本公开实施例还提供了一种软件包处理设备,该软件包处理设备可以包括处理器和存储器,存储器可以用于存储可执行指令。其中,处理器可以用于从存储器中读取可执行指令,并执行可执行指令以实现上述实施例中的软件包处理方法。

[0226] 图13示出了本公开实施例提供的一种软件包处理设备的结构示意图。下面具体参考图13,其示出了适于用来实现本公开实施例中的软件包处理设备1300的结构示意图。

[0227] 在本公开一些实施例中,软件包处理设备1300可以为图1中所示的软件包开发平台11,并执行上述的软件包开发方法。

[0228] 需要说明的是,图13示出的软件包处理设备1300仅仅是一个示例,不应对本公开实施例的功能和使用范围带来任何限制。

[0229] 如图13所示,该软件包处理设备1300可以包括处理装置(例如中央处理器、图形处理器等)1301,其可以根据存储在只读存储器(ROM)1302中的程序或者从存储装置1308加载到随机访问存储器(RAM)1303中的程序而执行各种适当的动作和处理。在RAM 1303中,还存储有软件包处理设备1300操作所需的各种程序和数据。处理装置1301、ROM 1302以及RAM 1303通过总线1304彼此相连。输入/输出(I/O)接口1305也连接至总线1304。

[0230] 通常,以下装置可以连接至I/O接口1305:包括例如触摸屏、触摸板、键盘、鼠标、摄像头、麦克风、加速度计、陀螺仪等的输入装置1306;包括例如液晶显示器(LCD)、扬声器、振动器等的输出装置13013;包括例如磁带、硬盘等的存储装置1308;以及通信装置1309。通信装置1309可以允许软件包处理设备1300与其他设备进行无线或有线通信以交换数据。虽然图13示出了具有各种装置的软件包处理设备1300,但是应理解的是,并不要求实施或具备所有示出的装置。可以替代地实施或具备更多或更少的装置。

[0231] 本公开实施例还提供了一种计算机可读存储介质,该存储介质存储有计算机程序,当计算机程序被处理器执行时,使得处理器实现上述实施例中的软件包处理方法。

[0232] 特别地,根据本公开的实施例,上文参考流程图描述的过程可以被实现为计算机软件程序。

[0233] 本公开实施例还提供了一种计算机程序产品,该计算机程序产品可以包括计算机程序,当计算机程序被处理器执行时,使得处理器实现上述实施例中的软件包处理方法。

[0234] 例如,本公开的实施例包括一种计算机程序产品,其包括承载在非暂态计算机可读介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中,该计算机程序可以通过通信装置1309从网络上被下载和安装,或者从存储装置1308被安装,或者从ROM 1302被安装。在该计算机程序被处理装置1301执行时,执行本公开实施例的软件包处理方法中限定的上述功能。

[0235] 需要说明的是,本公开上述的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或者是上述两者的任意组合。计算机可读存储介质例如可以是——但不限于——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可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(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。在本公开中,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。而在本公开中,计算机可读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,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读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,该计算机可读信号介质可以发送、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。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,包括但不限于:电线、光缆、RF(射频)等等,或者上述的任意合适的组合。

[0236] 在一些实施方式中,客户端、服务器可以利用诸如HTTP之类的任何当前已知或未来研发的网络协议进行通信,并且可以与任意形式或介质的数字数据通信(例如,通信网络)互连。通信网络的示例包括局域网(“LAN”),广域网(“WAN”),网际网(例如,互联网)以及端对端网络(例如,ad hoc端对端网络),以及任何当前已知或未来研发的网络。

[0237] 上述计算机可读介质可以是上述软件包处理设备中所包含的;也可以是单独存在,而未装配入该软件包处理设备中。

[0238] 上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被该软件包处理设备执行时,使得该软件包处理设备执行:

[0239] 从服务器下载第一待编译软件包;

[0240] 对第一待编译软件包进行编译,得到第一已编译软件包;

[0241] 将第一已编译软件包存储至软件包管理节点的目标软件包库内,目标软件包库为与目标应用平台相对应的软件包库,目标应用平台为对第一已编译软件包存在发布需求的应用平台。

[0242] 在本公开实施例中,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公开的操作的计算机程序代码,上述程序设计语言包括但不限于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”语言或类似的程序设计语言。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)。

[0243] 附图中的流程图和框图,图示了按照本公开各种实施例的系统、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,该模块、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,两个接连地表示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,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。

[0244] 描述于本公开实施例中涉及到的单元可以通过软件的方式实现,也可以通过硬

件的方式来实现。其中,单元的名称在某种情况下并不构成对该单元本身的限定。

[0245] 本文中以上描述的功能可以至少部分地由一个或多个硬件逻辑部件来执行。例如,非限制性地,可以使用的示范类型的硬件逻辑部件包括:现场可编程门阵列(FPGA)、专用集成电路(ASIC)、专用标准产品(ASSP)、片上系统(SOC)、复杂可编程逻辑设备(CPLD)等等。

[0246] 在本公开的上下文中,机器可读介质可以是有形的介质,其可以包含或存储以供指令执行系统、装置或设备使用或与指令执行系统、装置或设备结合地使用的程序。机器可读介质可以是机器可读信号介质或机器可读储存介质。机器可读介质可以包括但不限于电子的、磁性的、光学的、电磁的、红外的、或半导体系统、装置或设备,或者上述内容的任何合适组合。机器可读存储介质的更具体示例会包括基于一个或多个线的电气连接、便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM或快闪存储器)、光纤、便捷式紧凑盘只读存储器(CD-ROM)、光学储存设备、磁储存设备、或上述内容的任何合适组合。

[0247] 以上描述仅为本公开的较佳实施例以及对所运用技术原理的说明。本领域技术人员应当理解,本公开中所涉及的公开范围,并不限于上述技术特征的特定组合而成的技术方案,同时也应涵盖在不脱离上述公开构思的情况下,由上述技术特征或其等同特征进行任意组合而形成的其它技术方案。例如上述特征与本公开中公开的(但不限于)具有类似功能的技术特征进行互相替换而形成的技术方案。

[0248] 此外,虽然采用特定次序描绘了各操作,但是这不应理解为要求这些操作以所示出的特定次序或以顺序次序执行来执行。在一定环境下,多任务和并行处理可能是有利的。同样地,虽然在上面论述中包含了若干具体实现细节,但是这些不应被解释为对本公开的范围的限制。在单独的实施例的上下文中描述的某些特征还可以组合地实现在单个实施例中。相反地,在单个实施例的上下文中描述的各种特征也可以单独地或以任何合适的子组合的方式实现在多个实施例中。

[0249] 尽管已经采用特定于结构特征和/或方法逻辑动作的语言描述了本主题,但是应当理解所附权利要求书中所限定的主题未必局限于上面描述的特定特征或动作。相反,上面所描述的特定特征和动作仅仅是实现权利要求书的示例形式。

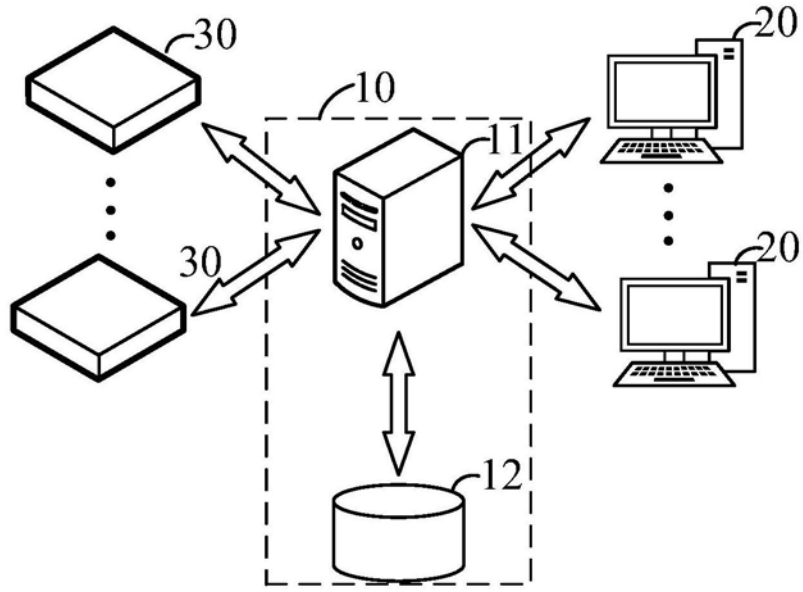


图1

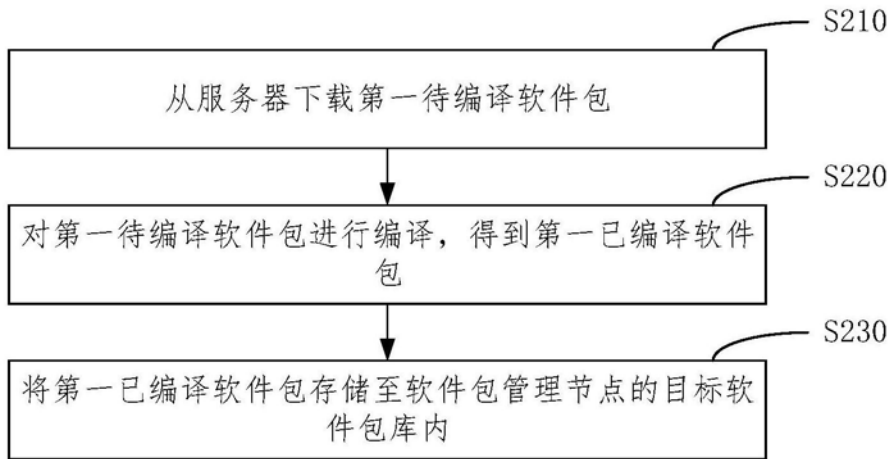


图2

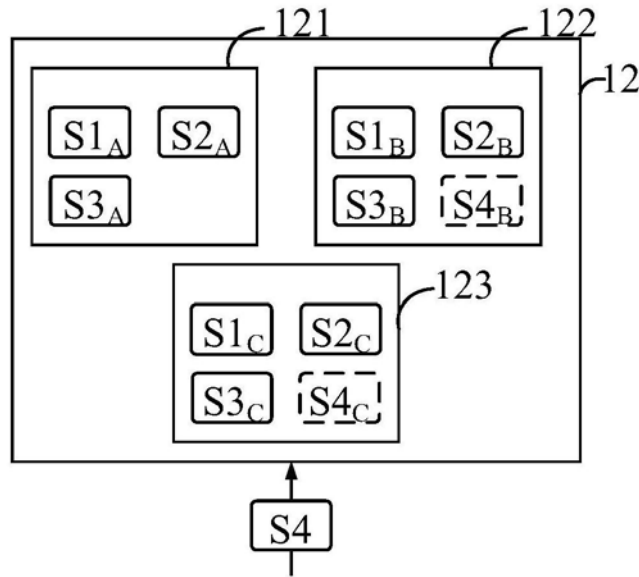


图3

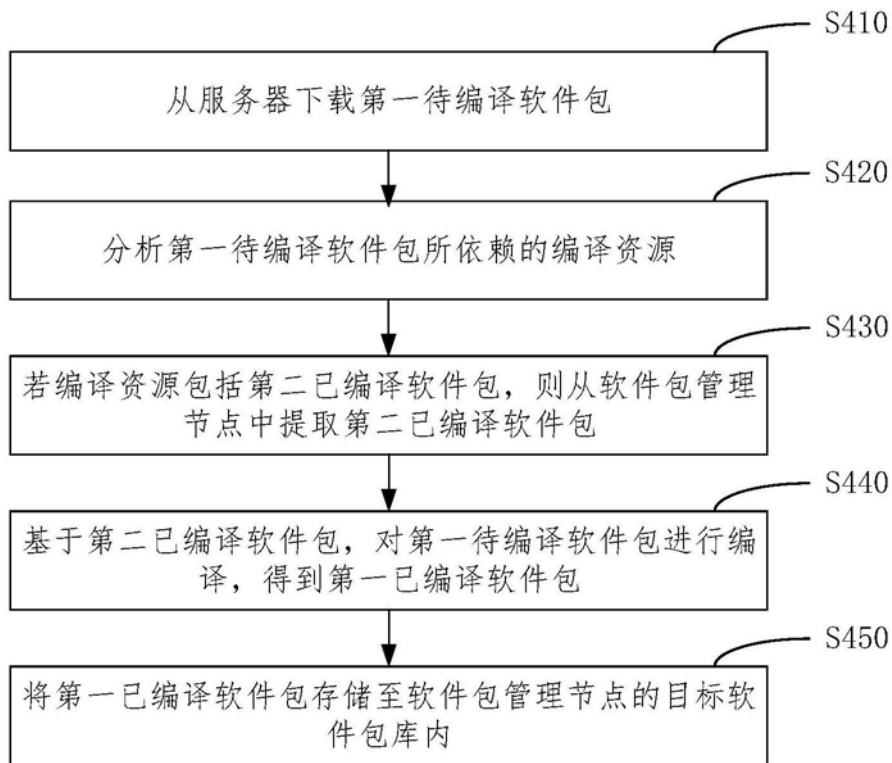


图4

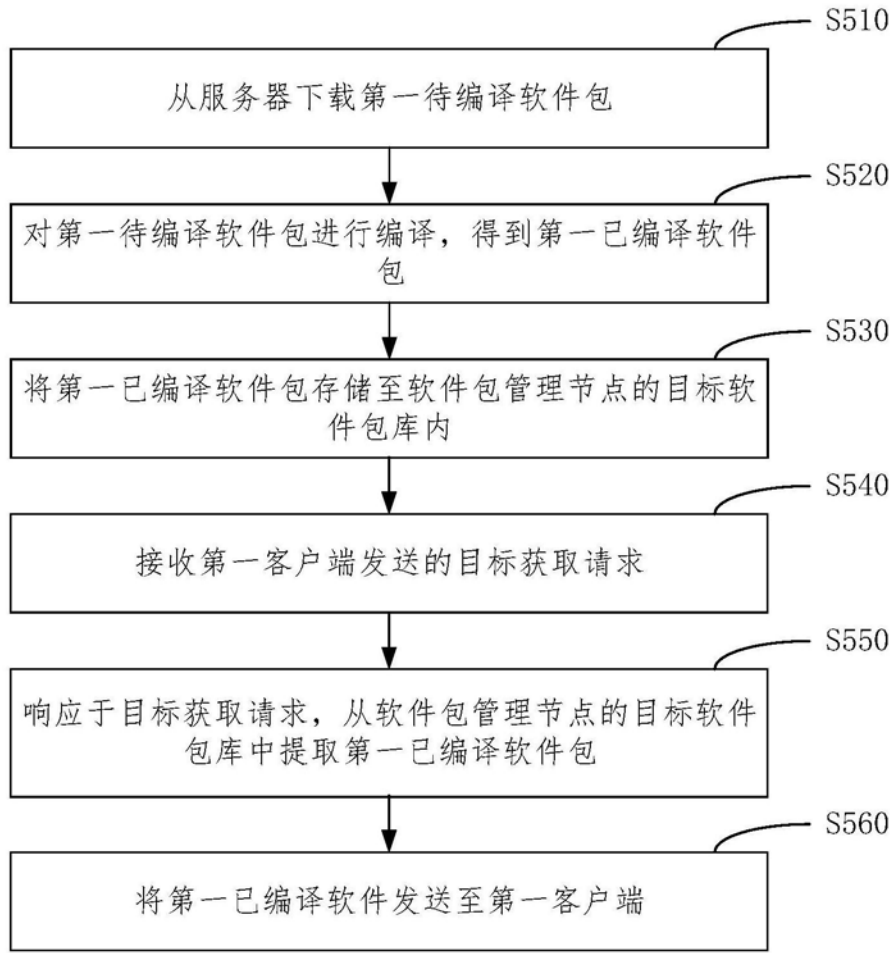


图5

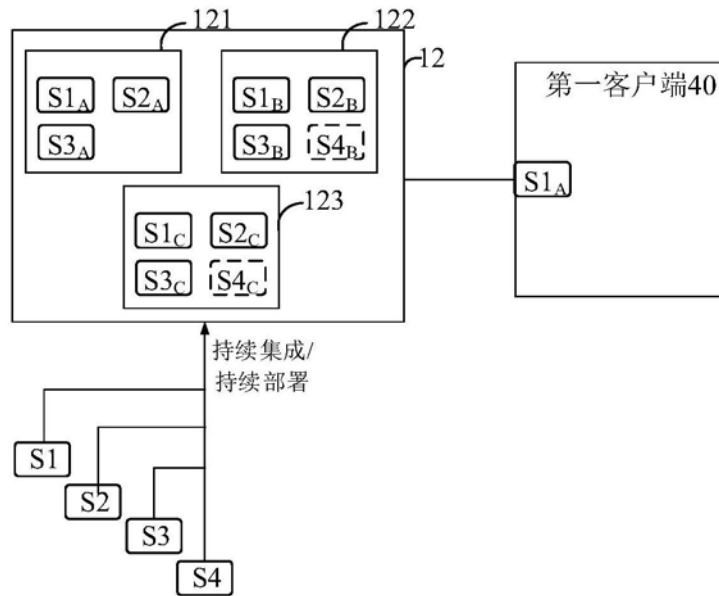


图6

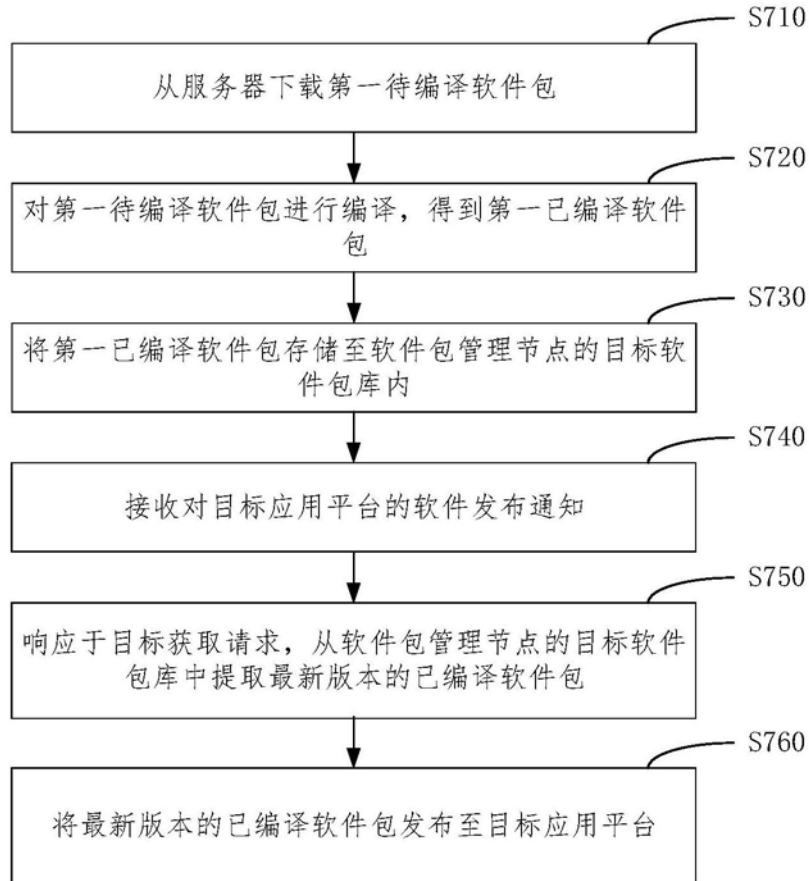


图7

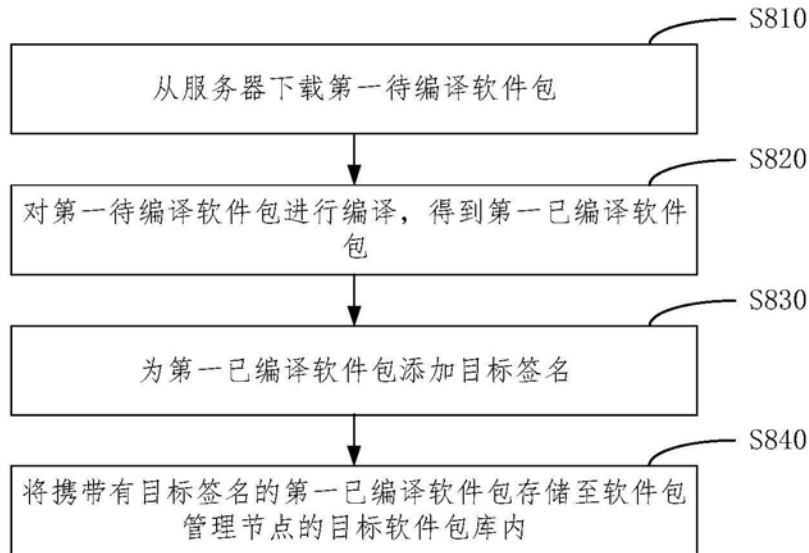


图8

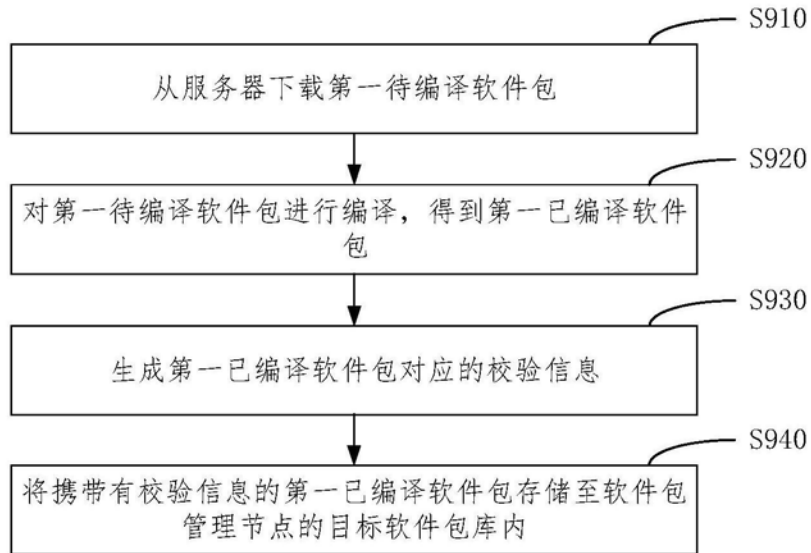


图9

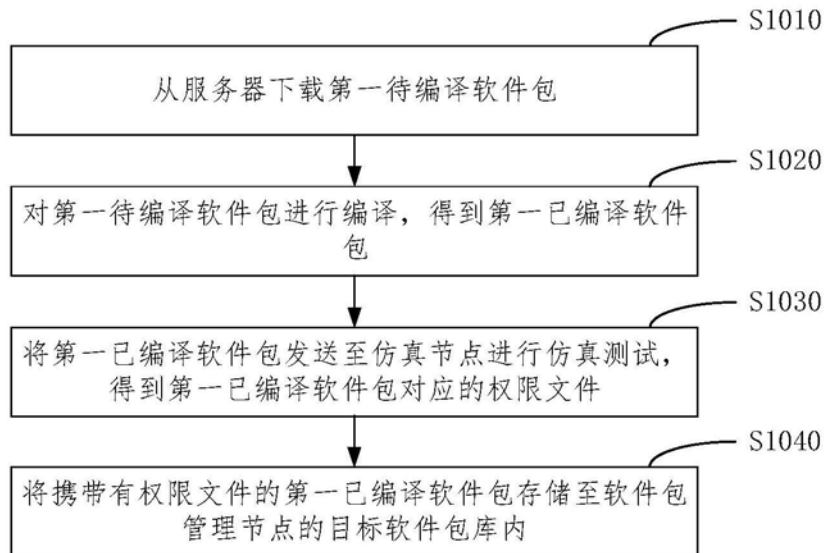


图10

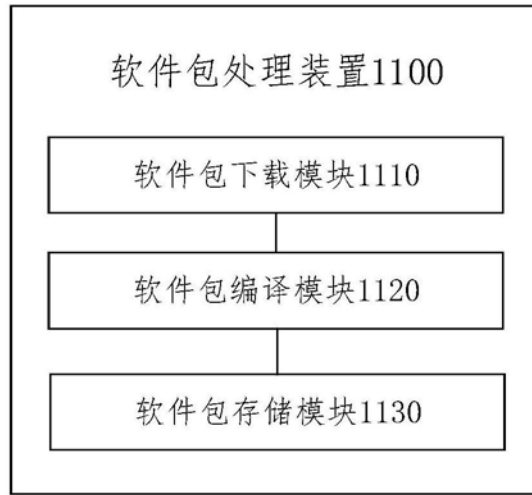


图11

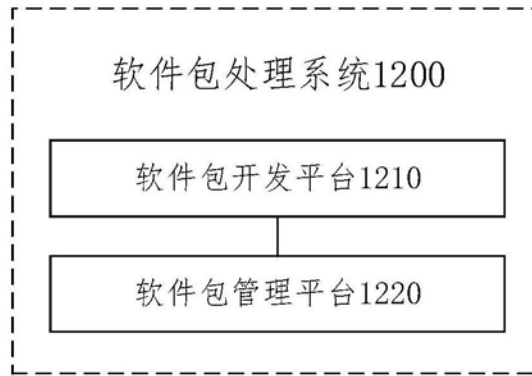


图12

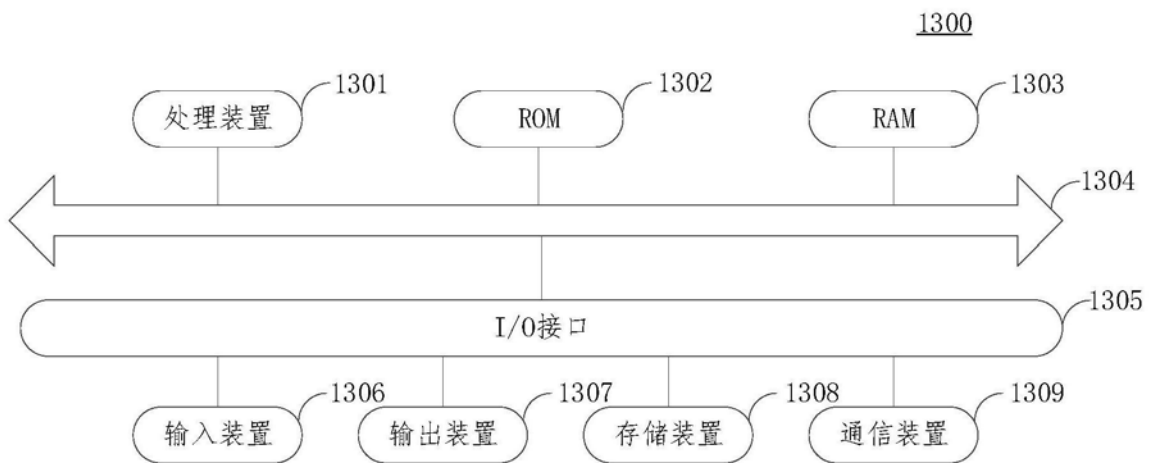


图13