



(12)发明专利申请

(10)申请公布号 CN 111552508 A

(43)申请公布日 2020.08.18

(21)申请号 202010357876.7

(22)申请日 2020.04.29

(71)申请人 杭州数梦工场科技有限公司
地址 310024 浙江省杭州市转塘科技经济
区块16号4幢326室

(72)发明人 陈军 高昕羽

(74)专利代理机构 北京博思佳知识产权代理有
限公司 11415

代理人 林祥

(51)Int.Cl.

G06F 8/71(2018.01)

G06F 8/41(2018.01)

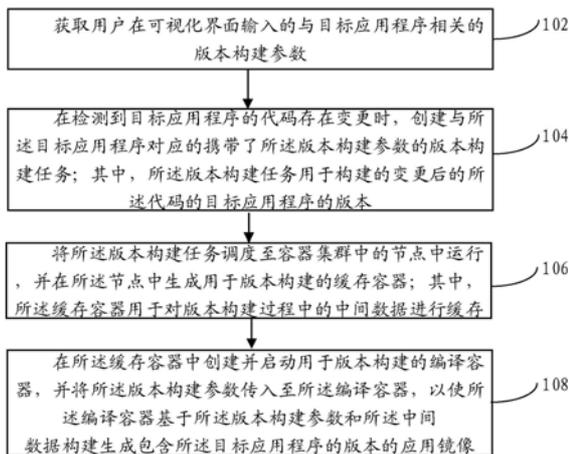
权利要求书2页 说明书12页 附图2页

(54)发明名称

应用程序版本构建方法、装置、电子设备

(57)摘要

公开一种应用程序版本构建方法、装置、电子设备及机器可读存储介质。在本申请中,获取用户在可视化界面输入的与目标应用程序相关的版本构建参数;在检测到目标应用程序的代码存在变更时,创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务;其中,所述版本构建任务用于构建的变更后的所述代码的目标应用程序的版本。将所述版本构建任务调度至容器集群中的节点中运行,并在所述节点中生成用于版本构建的缓存容器;其中,所述缓存容器用于对版本构建过程中的中间数据进行缓存;在所述缓存容器中创建并启动用于版本构建的编译容器,并将所述版本构建参数传入至所述编译容器,以使所述编译容器基于所述版本构建参数和所述中间数据构建生成包含所述目标应用程序的版本的版本的应用镜像。



1. 一种应用程序版本构建方法,所述方法包括:

获取用户在可视化界面输入的与目标应用程序相关的版本构建参数;

在检测到目标应用程序的代码存在变更时,创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务;其中,所述版本构建任务用于构建的变更后的所述代码的目标应用程序的版本;

将所述版本构建任务调度至容器集群中的节点中运行,并在所述节点中生成用于版本构建的缓存容器;其中,所述缓存容器用于对版本构建过程中的中间数据进行缓存;

在所述缓存容器中创建并启动用于版本构建的编译容器,并将所述版本构建参数传入至所述编译容器,以使所述编译容器基于所述版本构建参数和所述中间数据构建生成包含所述目标应用程序的版本的镜像。

2. 根据权利要求1所述的方法,在创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务之前,还包括:

检查与变更后的所述代码对应合入所述目标应用程序的代码版本分支是否为允许执行构建状态;

若是,则进一步创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务;否则,禁止版本构建任务创建。

3. 根据权利要求1所述的方法,所述版本构建参数包括用于生成所述缓存容器的第一容器镜像,所述在所述节点中生成用于版本构建的缓存容器,包括:

将所述第一容器镜像作为模板,在所述节点中生成所述第一容器镜像的第一容器实例,将所述第一容器实例作为用于版本构建的缓存容器。

4. 根据权利要求3所述的方法,所述版本构建参数还包括用于生成所述编译容器的第二容器镜像、用于编译目标应用程序的代码的编译环境和编译脚本;

所述在所述缓存容器中创建并启动用于版本构建的编译容器,并将所述版本构建参数传入至所述编译容器,包括:

将所述第二容器镜像作为模板,在所述缓存容器中生成所述第二容器镜像的第二容器实例,将所述第二容器实例作为用于版本构建的编译容器;

在所述编译容器启动运行后,将所述版本构建参数中的编译环境、编译脚本,作为参数传入至启动运行的所述编译容器。

5. 根据权利要求4所述的方法,所述版本构建参数还包括目标应用程序的代码获取地址,还包括:

所述编译容器从与基于所述代码获取地址对应的代码仓库中,获取目标应用程序的代码,并基于传入的所述版本构建参数中的编译环境、编译脚本,对所述代码执行版本编译,生成对应的目标应用程序的版本。

6. 根据权利要求5所述的方法,所述版本构建参数还包括作为目标应用程序的运行环境的第三容器镜像,还包括:

启动以所述第三容器镜像作为模板的第三容器实例;

所述编译容器将生成的目标应用程序的版本,部署至所述第三容器实例中,并生成所述第三容器实例作为模板的镜像;

并生成与所述镜像对应的应用镜像标签;其中,所述应用镜像标签至少包括目标

应用程序对应的代码版本分支标签和目标应用程序的代码提交标签。

7. 根据权利要求5所述的方法,还包括:

所述缓存容器将在所述编译容器生成目标应用程序的版本过程中所需的中间数据预先缓存在所述缓存容器本地的第一缓存目录下,以使所述编译容器基于所述缓存容器传入的环境变量,将所述第一缓存目录挂载到所述编译容器本地的第二缓存目录下,用于生成目标应用程序的版本。

8. 根据权利要求7所述的方法,所述中间数据至少包括容器镜像、版本编译依赖数据。

9. 根据权利要求8所述的方法,所述中间数据还包括版本编译过程的编译中间产物数据;在所述编译容器生成目标应用程序的版本过程中,还包括:

所述编译容器将所述编译中间产物数据保存至所述缓存容器中的第一缓存目录中进行缓存。

10. 根据权利要求1所述的方法,所述容器集群为基于Kubernetes管理的Docker容器集群。

11. 根据权利要求1所述的方法,所述缓存容器和所述编译容器,都为Docker容器。

12. 一种应用程序版本构建装置,所述装置包括:

获取模块,获取用户在可视化界面输入的与目标应用程序相关的版本构建参数;

创建模块,在检测到目标应用程序的代码存在变更时,创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务;其中,所述版本构建任务用于构建的变更后的所述代码的目标应用程序的版本;

生成模块,将所述版本构建任务调度至容器集群中的节点中运行,并在所述节点中生成用于版本构建的缓存容器;其中,所述缓存容器用于对版本构建过程中的中间数据进行缓存;

构建模块,在所述缓存容器中创建并启动用于版本构建的编译容器,并将所述版本构建参数传入至所述编译容器,以使所述编译容器基于所述版本构建参数和所述中间数据构建生成包含所述目标应用程序的版本的镜像。

13. 一种电子设备,包括通信接口、处理器、存储器和总线,所述通信接口、所述处理器和所述存储器之间通过总线相互连接;

所述存储器中存储机器可读指令,所述处理器通过调用所述机器可读指令,执行如权利要求1至11任一项所述的方法。

14. 一种机器可读存储介质,所述机器可读存储介质存储有机器可读指令,所述机器可读指令在被处理器调用和执行时,实现权利要求1至11任一项所述的方法。

应用程序版本构建方法、装置、电子设备

技术领域

[0001] 本申请一个或多个实施例涉及计算机应用技术领域,尤其涉及应用程序版本构建方法、装置、电子设备及机器可读存储介质。

背景技术

[0002] 持续集成,是指将应用程序中的由多人分工的多次开发的部分代码集成一起,形成应用程序的多个版本的主干代码,通过持续集成工具进行自动化方式的构建,编译和测试,提供可以部署发布的应用程序。持续集成的目的,就是让应用程序的版本开发和部署可以快速迭代,同时还能保持高质量。

发明内容

[0003] 本申请提供一种应用程序版本构建方法,所述方法包括:

[0004] 获取用户在可视化界面输入的与目标应用程序相关的版本构建参数;

[0005] 在检测到目标应用程序的代码存在变更时,创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务;其中,所述版本构建任务用于构建的变更后的所述代码的目标应用程序的版本;

[0006] 将所述版本构建任务调度至容器集群中的节点中运行,并在所述节点中生成用于版本构建的缓存容器;其中,所述缓存容器用于对版本构建过程中的中间数据进行缓存;

[0007] 在所述缓存容器中创建并启动用于版本构建的编译容器,并将所述版本构建参数传入至所述编译容器,以使所述编译容器基于所述版本构建参数和所述中间数据构建生成包含所述目标应用程序的版本的镜像。

[0008] 可选的,在创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务之前,还包括:

[0009] 检查与变更后的所述代码对应合入所述目标应用程序的代码版本分支是否为允许执行构建状态;

[0010] 若是,则进一步创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务;否则,禁止版本构建任务创建。

[0011] 可选的,所述版本构建参数包括用于生成所述缓存容器的第一容器镜像,所述在所述节点中生成用于版本构建的缓存容器,包括:

[0012] 将所述第一容器镜像作为模板,在所述节点中生成所述第一容器镜像的第一容器实例,将所述第一容器实例作为用于版本构建的缓存容器。

[0013] 可选的,所述版本构建参数还包括用于生成所述编译容器的第二容器镜像、用于编译目标应用程序的代码的编译环境和编译脚本;

[0014] 所述在所述缓存容器中创建并启动用于版本构建的编译容器,并将所述版本构建参数传入至所述编译容器,包括:

[0015] 将所述第二容器镜像作为模板,在所述缓存容器中生成所述第二容器镜像的第二

容器实例,将所述第二容器实例作为用于版本构建的编译容器;

[0016] 在所述编译容器启动运行后,将所述版本构建参数中的编译环境、编译脚本,作为参数传入至启动运行的所述编译容器。

[0017] 可选的,所述版本构建参数还包括目标应用程序的代码获取地址,还包括:

[0018] 所述编译容器从与基于所述代码获取地址对应的代码仓库中,获取目标应用程序的代码,并基于传入的所述版本构建参数中的编译环境、编译脚本,对所述代码执行版本编译,生成对应的目标应用程序的版本。

[0019] 可选的,所述版本构建参数还包括作为目标应用程序的运行环境的第三容器镜像,还包括:

[0020] 启动以所述第三容器镜像作为模板的第三容器实例;

[0021] 所述编译容器将生成的目标应用程序的版本,部署至所述第三容器实例中,并生成所述第三容器实例作为模板的应用镜像;

[0022] 并生成与所述应用镜像对应的应用镜像标签;其中,所述应用镜像标签至少包括目标应用程序对应的代码版本分支标签和目标应用程序的代码提交标签。

[0023] 可选的,还包括:

[0024] 所述缓存容器将在所述编译容器生成目标应用程序的版本过程中所需的中间数据预先缓存在所述缓存容器本地的第一缓存目录下,以使所述编译容器基于所述缓存容器传入的环境变量,将所述第一缓存目录挂载到所述编译容器本地的第二缓存目录下,用于生成目标应用程序的版本。

[0025] 可选的,所述中间数据至少包括容器镜像、版本编译依赖数据。

[0026] 可选的,所述中间数据还包括版本编译过程的编译中间产物数据;在所述编译容器生成目标应用程序的版本过程中,还包括:

[0027] 所述编译容器将所述编译中间产物数据保存至所述缓存容器中的第一缓存目录中进行缓存。

[0028] 可选的,所述容器集群为基于Kubernetes管理的Docker容器集群。

[0029] 可选的,所述缓存容器和所述编译容器,都为Docker容器。

[0030] 本申请还提供一种应用程序版本构建装置,所述装置包括:

[0031] 获取模块,获取用户在可视化界面输入的与目标应用程序相关的版本构建参数;

[0032] 创建模块,在检测到目标应用程序的代码存在变更时,创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务;其中,所述版本构建任务用于构建的变更后的所述代码的目标应用程序的版本;

[0033] 生成模块,将所述版本构建任务调度至容器集群中的节点中运行,并在所述节点中生成用于版本构建的缓存容器;其中,所述缓存容器用于对版本构建过程中的中间数据进行缓存;

[0034] 构建模块,在所述缓存容器中创建并启动用于版本构建的编译容器,并将所述版本构建参数传入至所述编译容器,以使所述编译容器基于所述版本构建参数和所述中间数据构建生成包含所述目标应用程序的版本的的应用镜像。

[0035] 可选的,在创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务之前,还包括:

[0036] 检查模块,检查与变更后的所述代码对应合入所述目标应用程序的代码版本分支是否为允许执行构建状态;

[0037] 若是,则进一步创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务;否则,禁止版本构建任务创建。

[0038] 可选的,所述版本构建参数包括用于生成所述缓存容器的第一容器镜像,所述生成模块进一步:

[0039] 将所述第一容器镜像作为模板,在所述节点中生成所述第一容器镜像的第一容器实例,将所述第一容器实例作为用于版本构建的缓存容器。

[0040] 可选的,所述版本构建参数还包括用于生成所述编译容器的第二容器镜像、用于编译目标应用程序的代码的编译环境和编译脚本;

[0041] 所述构建模块进一步:

[0042] 将所述第二容器镜像作为模板,在所述缓存容器中生成所述第二容器镜像的第二容器实例,将所述第二容器实例作为用于版本构建的编译容器;

[0043] 在所述编译容器启动运行后,将所述版本构建参数中的编译环境、编译脚本,作为参数传入至启动运行的所述编译容器。

[0044] 可选的,所述版本构建参数还包括目标应用程序的代码获取地址,所述构建模块进一步:

[0045] 所述编译容器从与基于所述代码获取地址对应的代码仓库中,获取目标应用程序的代码,并基于传入的所述版本构建参数中的编译环境、编译脚本,对所述代码执行版本编译,生成对应的目标应用程序的版本。

[0046] 可选的,所述版本构建参数还包括作为目标应用程序的运行环境的第三容器镜像,所述构建模块进一步:

[0047] 启动以所述第三容器镜像作为模板的第三容器实例;

[0048] 所述编译容器将生成的目标应用程序的版本,部署至所述第三容器实例中,并生成所述第三容器实例作为模板的应用镜像;

[0049] 并生成与所述应用镜像对应的应用镜像标签;其中,所述应用镜像标签至少包括目标应用程序对应的代码版本分支标签和目标应用程序的代码提交标签。

[0050] 可选的,还包括:

[0051] 缓存模块,所述缓存容器将在所述编译容器生成目标应用程序的版本过程中所需的中间数据预先缓存在所述缓存容器本地的第一缓存目录下,以使所述编译容器基于所述缓存容器传入的环境变量,将所述第一缓存目录挂载到所述编译容器本地的第二缓存目录下,用于生成目标应用程序的版本。

[0052] 可选的,所述中间数据至少包括容器镜像、版本编译依赖数据。

[0053] 可选的,所述中间数据还包括版本编译过程的编译中间产物数据;在所述编译容器生成目标应用程序的版本过程中,所述缓存模块进一步:

[0054] 所述编译容器将所述编译中间产物数据保存至所述缓存容器中的第一缓存目录中进行缓存。

[0055] 可选的,所述容器集群为基于Kubernetes管理的Docker容器集群。

[0056] 可选的,所述缓存容器和所述编译容器,都为Docker容器。

[0057] 本申请还提供一种电子设备,包括通信接口、处理器、存储器和总线,所述通信接口、所述处理器和所述存储器之间通过总线相互连接;

[0058] 所述存储器中存储机器可读指令,所述处理器通过调用所述机器可读指令,执行上述的方法。

[0059] 通过以上实施例,通过将目标应用程序相关的版本构建参数模板化并以可视化界面方式提供给用户进行版本构建,并在容器集群中构建创建多级容器进行版本构建以及对版本构建过程中的中间数据缓存,生成包含目标应用程序的版本的的应用镜像。一方面,降低了开发人员进行版本构建的学习成本,提高了版本开发效率;另一方面,保证了版本并发构建的隔离性,同时由于对版本构建过程中的中间数据缓存,避免了每次版本构建需要重新下载构建依赖,提高了版本构建效率。

附图说明

[0060] 图1是一示例性实施例提供的一种应用程序版本构建方法的流程图;

[0061] 图2是一示例性实施例提供的一种版本构建的多级容器嵌套的架构示意图;

[0062] 图3是一示例性实施例提供的一种电子设备的硬件结构图;

[0063] 图4是一示例性实施例提供的一种应用程序版本构建装置的框图。

具体实施方式

[0064] 这里将详细地对示例性实施例进行说明,其示例表示在附图中。下面的描述涉及附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本申请相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本申请的一些方面相一致的装置和方法的例子。

[0065] 在本申请使用的术语是仅仅出于描述特定实施例的目的,而非旨在限制本申请。在本申请和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式,除非上下文清楚地表示其他含义。还应当理解,本文中使用的术语“和/或”是指并包含一个或多个相关联的列出项目的任何或所有可能组合。

[0066] 应当理解,尽管在本申请可能采用术语第一、第二、第三等来描述各种信息,但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如,在不脱离本申请范围的情况下,第一信息也可以被称为第二信息,类似地,第二信息也可以被称为第一信息。取决于语境,如在此所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”。

[0067] 为了使本技术领域的人员更好地理解本说明书实施例中的技术方案,下面先对本说明书实施例涉及的应用程序版本构建的相关技术,进行简要说明。

[0068] 传统的持续集成通常基于Jenkins等持续集成工具实现,此类持续集成工具在版本构建时,普遍存在配置方式复杂(比如:需要为不同的语言框架手动下载、安装及配置对应的编译脚本、编译工具等)、版本构建部署在物理节点可扩展性差、单个物理节点上进行多个版本并发构建的隔离性差及代码残留安全隐患的问题。

[0069] 基于此,本申请旨在提出一种,基于将版本构建任务调度至容器集群节点的多级嵌套的容器中进行版本构建,并对版本构建过程中的中间数据进行缓存,从而生成包含目

标应用程序的版本的镜像的技术方案。

[0070] 在实现时,版本构建系统获取用户在可视化界面输入的与目标应用程序相关的版本构建参数;在检测到目标应用程序的代码存在变更时,创建与目标应用程序对应的携带了上述版本构建参数的版本构建任务;其中,上述版本构建任务用于构建的变更后的代码的目标应用程序的版本。

[0071] 进一步地,将上述版本构建任务调度至容器集群中的节点中运行,并在上述节点中生成用于版本构建的缓存容器;其中,缓存容器用于对版本构建过程中的中间数据进行缓存;在上述缓存容器中创建并启动用于版本构建的编译容器,并将上述版本构建参数传入至所述编译容器,以使上述编译容器基于上述版本构建参数和上述中间数据构建生成包含上述目标应用程序的版本的镜像。

[0072] 在以上方案中,通过将目标应用程序相关的版本构建参数模板化并以可视化界面方式提供给用户进行版本构建,并在容器集群中构建创建多级容器进行版本构建以及对版本构建过程的中间数据缓存,生成包含目标应用程序的版本的镜像。一方面,降低了开发人员进行版本构建的学习成本,提高了版本开发效率;另一方面,保证了版本并发构建的隔离性,同时由于对版本构建过程的中间数据缓存,避免了每次版本构建需要重新下载构建依赖,提高了版本构建效率。

[0073] 下面通过具体实施例并结合具体的应用场景对本申请进行描述。

[0074] 请参见图1,图1是本申请一实施例提供的一种应用程序版本构建方法的流程图,所述方法应用于版本构建系统,上述方法执行以下步骤:

[0075] 步骤102、获取用户在可视化界面输入的与目标应用程序相关的版本构建参数。

[0076] 步骤104、在检测到目标应用程序的代码存在变更时,创建与目标应用程序对应的携带了所述版本构建参数的版本构建任务;其中,所述版本构建任务用于构建的变更后的所述代码的目标应用程序的版本。

[0077] 步骤106、将所述版本构建任务调度至容器集群中的节点中运行,并在所述节点中生成用于版本构建的缓存容器;其中,缓存容器用于对版本构建过程中的中间数据进行缓存。

[0078] 步骤108、在所述缓存容器中创建并启动用于版本构建的编译容器,并将所述版本构建参数传入至所述编译容器,以使所述编译容器基于所述版本构建参数和所述中间数据构建生成包含所述目标应用程序的版本的镜像。

[0079] 在本说明书中,上述目标应用程序,可以包括基于任何编程语言及其相关框架开发的应用程序。

[0080] 例如,在实际应用中,上述目标应用程序,具体可以为基于Java语言开发的基于不同JDK(JavaDevelopmentKit,Java开发工具包)框架的Java应用程序。上述目标应用程序,也可以为基于.NET语言开发的基于不同.NET Framework 4.0开发框架的.NET应用程序。

[0081] 在本说明书中,上述版本构建系统,是指用于对上述目标应用程序的代码进行版本构建,生成不同版本的上述目标应用程序的节点或节点集群;

[0082] 其中,上述节点具体可以是由若干物理机器搭建的集群中的单个物理节点,也可以是若干虚拟机搭建的集群中的单个虚拟机节点

[0083] 例如,在实际应用中,上述版本构建系统具体可以是基于若干服务器硬件搭建而

成,也可以是基于若干KVM(Kernel-Based Virtual Machines,基于内核的虚拟机)虚拟机搭建而成。

[0084] 需要说明的是,当上述版本构建系统基于虚拟机节点构建时,可以利用虚拟机集群管理架构(比如:Openstack架构)对虚拟机节点进行动态弹性扩展。

[0085] 在本说明书中,上述容器集群,是指上述版本构建系统中包括的用于进行容器管理的集群。

[0086] 在示出的一种实施方式中,上述容器集群为基于Kubernetes(容器编排引擎,支持自动化部署、大规模可伸缩、应用容器化管理)管理的Docker容器集群;其中,上述容器集群具体可以部署在上述版本构建系统的物理节点或者虚拟机节点上。

[0087] 需要说明的是,上述容器集群除了可以管理Docker容器外,也可以管理其他容器,比如:LXC容器(Linux Container)。

[0088] 在本说明书中,上述版本构建参数,包括与构建上述目标程序相关的任何参数。例如,在实际应用中,上述版本构建参数具体可包括上述目标程序相关的代码、代码分支、编译环境和编译脚本等信息。

[0089] 在本说明书中,上述版本构建系统可以向用户(比如:上述目标程序的开发人员、版本管理人员、测试人员等)输出可视化界面,以使用户在该可视化界面选择用于构建上述目标应用程序的上述版本构建参数,无需用户熟悉持续集成工具并进行复杂配置(比如:需要为不同的语言框架手动下载、安装及配置对应的编译脚本、编译工具等),由此降低了开发人员进行版本构建的学习成本,提高了版本开发效率。

[0090] 在本说明书中,在获取用户在可视化界面输入的与上述目标应用程序相关的上述版本构建参数后,上述版本构建系统可以基于上述版本构建参数进行上述目标应用程序的版本构建。

[0091] 在本说明书中,上述版本构建系统可以基于上述版本构建系统中包括的代码仓库服务,检测用户提交的上述目标应用程序的代码是否发生变化。

[0092] 在示出的一种实施方式中,在监测到上述目标应用程序的代码存在变更时,上述版本构建系统可以进一步检查与变更后的上述代码对应合入上述目标应用程序的代码版本分支是否为允许执行构建状态,若是,则进一步创建与上述目标应用程序对应的携带了上述版本构建参数的版本构建任务;否则,禁止版本构建任务创建。

[0093] 例如,在实际应用中,上述目标应用程序的代码对应合入的代码版本分支未通过自动化测试验证,则上述版本构建系统可以获取该代码版本分支的自动化测试验证的失败结果,从而根据该失败结果确定该代码版本分支为禁止执行构建状态;而当上述目标应用程序的代码对应合入的代码版本分支通过自动化测试验证,则上述版本构建系统可以获取该代码版本分支的自动化测试验证的成功结果,从而根据该成功结果确定该代码版本分支为允许执行构建状态。

[0094] 本说明书中,上述编译容器,是指用于上述目标应用程序的代码编译,生成包含上述目标应用程序版本的应用镜像镜像的容器。例如,在示出的一种实施方式中,上述编译容器为Docker容器。

[0095] 本说明书中,上述缓存容器,是指用于对上述编译容器执行版本构建过程中的中间数据进行缓存的容器。例如,在示出的一种实施方式中,上述缓存容器为Docker容器,该

Docker容器用于对上述编译容器执行版本构建过程中的中间数据进行缓存。

[0096] 本说明书中,上述中间数据,是指上述编译容器在上述目标应用程序的版本构建过程中所需要的、除上述目标应用程序的代码以外的编译依赖工具、容器镜像等。

[0097] 在本说明书中,在确定变更后的上述目标应用程序的代码对应合入代码版本分支为允许执行构建状态后,上述版本构建系统创建与上述目标应用程序对应的携带了上述版本构建参数的版本构建任务;其中,上述版本构建任务用于构建的变更后的上述代码的目标应用程序的版本。

[0098] 在本说明书中,进一步地,上述版本构建系统可以将上述版本构建任务调度至上述容器集群中的节点中运行,并在该节点中生成用于版本构建的缓存容器;

[0099] 例如,上述版本构建系统可以将版本构建任务以Job方式,调度至基于Kubernetes管理的Docker容器集群中的虚拟机节点上运行,并在该虚拟机节点上生成上述缓存容器。

[0100] 需要说明的是,基于Job方式实现版本构建任务,在Job完成后就会自动退出,可以保证每次构建任务完成后该任务创建的构建环境都会自动销毁,不会保留代码等敏感信息,Job方式的具体过程及实现请参见Kubernetes相关的技术文档,这里不再赘述。

[0101] 在示出的一种实施方式中,上述版本构建参数包括用于生成上述缓存容器的第一容器镜像,在上述节点中生成上述缓存容器的过程中,上述版本构建系统可以将上述第一容器镜像作为模板,在上述节点中生成上述第一容器镜像的第一容器实例,将上述第一容器实例作为用于版本构建的上述缓存容器。

[0102] 例如,在实际应用中,上述版本构建参数包括用于生成上述缓存容器的linux容器镜像(第一容器镜像),上述版本构建系统可以将上述第一容器镜像作为模板,在上述节点中生成linux容器镜像的linux容器实例,作为用于版本构建的上述缓存容器。

[0103] 在本说明书中,在上述缓存容器生成后,上述版本构建系统在上述缓存容器中创建并启动用于版本构建的上述编译容器。

[0104] 在示出的一种实施方式中,上述版本构建参数除了包括上述第一容器镜像外,还包括用于生成上述编译容器的第二容器镜像、用于编译目标应用程序的代码的编译环境和编译脚本,上述版本构建系统将上述第二容器镜像作为模板,在上述缓存容器中生成上述第二容器镜像的第二容器实例,将上述第二容器实例作为用于版本构建的编译容器。

[0105] 接着以上示例继续举例,上述版本构建参数包括用于生成上述编译容器的JDK容器镜像(第二容器镜像),上述版本构建系统在生成linux容器实例作为上述缓存容器后,在该linux容器中生成JDK容器镜像的第二容器实例,将JDK容器镜像实例作为用于版本构建的上述编译容器。

[0106] 需要说明的是,在上述缓存容器中,可以创建用于上述目标程序版本构建的不同编程语言及其依赖的框架、工具版本对应的多个上述编译容器。

[0107] 例如,以上述目标程序版本为JAVA应用程序为例,对应JDK版本存在1.4、1.6多个版本,则在上述缓存容器中可以创建与JDK1.4、JDK1.6分别对应的两个编译容器。

[0108] 又例如,以上述目标程序版本可以基于JAVA语言或C++语言开发,则在上述缓存容器中可以创建与JAVA语言依赖的框架及JAVA编译工具对应的编译容器,或者与C++语言依赖的框架及C++编译工具对应的编译容器。

[0109] 在本说明书中,在上述编译容器生成后,上述缓存容器可以将上述版本构建参数

中的用于编译目标应用程序的代码的编译环境和编译脚本传入至上述编译容器,以使上述编译容器基于上述版本构建参数和上述中间数据构建生成包含上述目标应用程序的版本的镜像。

[0110] 例如,以上述目标程序版本以基于JAVA语言开发为例,上述缓存容器可以将版本构建参数中的用于编译目标应用程序的代码的JAVA编译环境(比如:JDK1.8)和编译脚本(比如:maven编译脚本,类似`mvn clean package-DskipTests=true echo`)传入至上述编译容器,以使上述编译容器基于上述版本构建参数和上述中间数据构建生成包含上述目标应用程序的版本的镜像。

[0111] 在示出的一种实施方式中,上述版本构建参数除了包括上述第一容器镜像、用于生成上述编译容器的第二容器镜像、用于编译目标应用程序的代码的编译环境和编译脚本外,还包括上述目标应用程序的代码获取地址,上述编译容器从与基于上述代码获取地址对应的代码仓库中,获取上述目标应用程序的代码,并基于传入的上述版本构建参数中的编译环境、编译脚本,对上述代码执行版本编译,生成对应的目标应用程序的版本。

[0112] 例如,上述版本构建参数除了包括上述第一容器镜像、用于生成上述编译容器的第二容器镜像、用于编译目标应用程序的代码的编译环境和编译脚本外,还包括上述目标应用程序的代码获取地址;其中,上述目标应用程序的代码获取地址具体可以是位于公网或私网的代码仓库中的目标应用程序的相关代码的代码获取地址,上述编译容器从与基于上述代码获取地址对应的代码仓库中,获取目标应用程序的代码,并基于传入的上述版本构建参数中的编译环境、编译脚本,对该代码执行版本编译,生成对应的目标应用程序的版本。

[0113] 需要说明的是,上述代码仓库具体可以包括github代码仓库、svn代码仓库、cvs代码仓库等,上述代码仓库具体可以部署在私网内或公网中,上述代码仓库的具体类型和部署方式,在本说明书中不作具体限定。

[0114] 在示出的一种实施方式中,上述版本构建参数除了包括上述第一容器镜像、用于生成上述编译容器的第二容器镜像、用于编译目标应用程序的代码的编译环境和编译脚本、上述目标应用程序的代码获取地址外,还包括作为上述目标应用程序的运行环境的第三容器镜像,上述编译容器启动以上述第三容器镜像作为模板的第三容器实例,将生成的目标应用程序的版本,部署至上述第三容器实例中,并生成上述第三容器实例作为模板的镜像;并生成与上述应用镜像对应的应用镜像标签;其中,上述应用镜像标签至少包括上述目标应用程序对应的代码版本分支标签和上述目标应用程序的代码提交标签。

[0115] 例如,上述版本构建参数除了包括上述第一容器镜像、用于生成上述编译容器的第二容器镜像、用于编译目标应用程序的代码的编译环境和编译脚本、上述目标应用程序的代码获取地址外,还包括作为上述目标应用程序的运行环境的第三容器镜像;其中,上述第三容器镜像具体可以包括上述目标应用程序将要部署的运行环境,比如:上述目标应用程序为JAVA程序,则上述第三容器镜像可以为基于包括JAVA程序运行环境JRE的linux容器的容器镜像。上述编译容器启动以linux容器的容器镜像作为模板的linux容器实例,将生成的JAVA程序的版本,部署至该linux容器实例中,并生成该linux容器实例作为模板的镜像,同时生成与上述应用镜像对应的应用镜像标签;其中,该应用镜像标签至少包括JAVA程序对应的代码版本分支标签和JAVA程序的代码提交标签。

[0116] 需要说明的是,上述代码版本分支标签,是指上述目标应用程序对应的代码版本分支的标识,通过在该应用镜像标签中记录代码版本分支标签,可以方便用户明确区分上述目标应用程序的代码合入版本,比如:主线版本或分支版本。上述代码提交标签,是指上述目标应用程序开发人员提交的修改代码对应的修改标识,该修改标识通常也被称为 commitid,通过在该应用镜像标签中记录上述代码提交标签,可以方便用户明确代码修改的具体内容、原因、时间、修改人等信息,从而方便用户进行版本回溯。

[0117] 当然,在实际应用中,上述应用镜像标签除了包括上述码版本分支标签和上述代码提交标签外,还可以包括上述应用镜像的操作系统类型(比如:linux及详细linux发行版本)、容器引擎版本(比如:Docker版本)、运行硬件架构(比如:AMD架构、Intel架构、各种嵌入式架构)、镜像摘要(上述应用镜像经哈希算法计算得到的哈希值)。

[0118] 在示出的一种实施方式中,在上述编译容器基于上述版本构建参数和上述中间数据构建生成包含上述目标应用程序的版本的上述应用镜像的过程中,上述缓存容器将在上述编译容器生成上述目标应用程序的版本过程中所需的中间数据预先缓存在上述缓存容器本地的第一缓存目录下,以使上述编译容器基于上述缓存容器传入的环境变量,将上述第一缓存目录挂载到上述编译容器本地的第二缓存目录下,用于生成上述目标应用程序的版本;其中,上述中间数据至少包括容器镜像、版本编译依赖数据。

[0119] 例如,以上述目标应用程序为将部署运行在linux系统的、基于JRE1.8的JAVA程序为例,上述缓存容器将在上述编译容器生JAVA程序的版本过程中所需的linux容器镜像、版本编译依赖的JRE1.8安装包及其它编译依赖工具等中间数据,预先缓存在上述缓存容器本地的第一缓存目录下,并通过shell环境变量的方式将第一缓存目录传给上述编译容器,上述编译容器可以基于上述缓存容器传入的环境变量获取上述缓存容器本地的第一缓存目录的信息,进一步基于容器加载命令,将上述第一缓存目录挂载到上述编译容器本地的第二缓存目录下,从而使得上述编译容器可以远程访问上述第一缓存目录下的上述中间数据,将上述中间数据拷贝到上述编译容器本地,用于执行目标应用程序的版本编译。

[0120] 需要说明的是,基于上述缓存容器将在上述编译容器生成上述目标应用程序的版本过程中所需的中间数据预先缓存在上述缓存容器本地的第一缓存目录,可以在编译容器中对上述目标程序进行每次版本构建时,需要重复下载构建依赖,提高了上述目标应用程序的构建效率。

[0121] 在示出的一种实施方式中,上述中间数据除了包括容器镜像、版本编译依赖数据外,还可以包括上述编译容器在版本编译过程的编译中间产物数据,在上述编译容器构建生成包含上述目标应用程序的版本的上述应用镜像的过程中,上述编译容器可以将上述编译中间产物数据,保存至上述缓存容器中的第一缓存目录中进行缓存。

[0122] 接着以上示例继续举例,上述编译容器可以将编译JAVA程序的编译中间产物数据,比如:编译JAVA程序中相关的JAVA类库的字节码,保存至上述缓存容器中的第一缓存目录中进行缓存。

[0123] 需要说明的是,基于上述编译容器可将上述编译中间产物数据保存至上述缓存容器中的第一缓存目录中进行缓存,可以减少上述目标应用程序的多次版本构建时对依赖的类库重复进行编译,提高了上述目标应用程序的构建效率。

[0124] 为了方便理解缓存容器和编译容器的关系。请参见图2,图2是一示例性实施例提

供的一种版本构建的多级容器嵌套的架构示意图。

[0125] 如图2所示,缓存容器运行在容器集群中的节点之上,缓存容器中包括/data/bulidcache目录(第一缓存目录),缓存容器中可以嵌套运行有构建任务A触发创建的编译容器A、构建任务B触发创建的编译容器B;其中,编译容器A中包括/data/bulid1目录(第二缓存目录)、编译容器B中包括/data/bulid2目录(第二缓存目录),编译容器A、B可以以目录挂载方式将缓存容器中的/data/bulidcache目录分别加载到各自的/data/bulid1目录和/data/bulid2目录,进行上述中间数据(比如:构建目标程序所需依赖的容器镜像、版本编译依赖数据等通用组件、构建目标程序生成的编译中间产物等)的获取或保存。

[0126] 如图2所示,编译容器A、B可以同时并发执行多个目标程序的版本编译及其对应的应用镜像生成。

[0127] 在本说明书中,在上述编译容器基构建生成包含上述目标应用程序的版本的镜像后,上述编译容器可以进一步将上述应用镜像上传到镜像仓库中进行永久保存。

[0128] 在本说明书中,进一步地,在上述版本构建任务完成后,上述版本构建任务将执行自动销毁处理,也即,上述版本构建系统将上述编译容器进行销毁上述编译容器中代码等敏感信息不会被保留。

[0129] 在以上技术方案中,通过将目标应用程序相关的版本构建参数模板化并以可视化界面方式提供给用户进行版本构建,并在容器集群中构建创建多级容器进行版本构建以及对版本构建过程的中间数据缓存,生成包含目标应用程序的版本的镜像。一方面,降低了开发人员进行版本构建的学习成本,提高了版本开发效率;另一方面,保证了版本并发构建的隔离性,同时由于对版本构建过程中的数据缓存,避免了每次版本构建需要重新下载构建依赖,提高了版本构建效率。

[0130] 与上述方法实施例相对应,本申请还提供了应用程序版本构建装置的实施例。

[0131] 与上述方法实施例相对应,本说明书还提供了一种应用程序版本构建装置的实施例。本说明书的应用程序版本构建装置的实施例可以应用在电子设备上。装置实施例可以通过软件实现,也可以通过硬件或者软硬件结合的方式实现。以软件实现为例,作为一个逻辑意义上的装置,是通过其所在电子设备的处理器将非易失性存储器中对应的计算机程序指令读取到内存中运行形成的。从硬件层面而言,如图3所示,为本说明书的应用程序版本构建装置所在电子设备的一种硬件结构图,除了图3所示的处理器、内存、网络接口、以及非易失性存储器之外,实施例中装置所在的电子设备通常根据该电子设备的实际功能,还可以包括其他硬件,对此不再赘述。

[0132] 图4是本说明书一示例性实施例示出的一种应用程序版本构建装置的框图。

[0133] 请参见图4,所述应用程序版本构建装置40可以应用在前述图3所示的电子设备中,所述装置包括:

[0134] 获取模块401,获取用户在可视化界面输入的与目标应用程序相关的版本构建参数;

[0135] 创建模块402,在检测到目标应用程序的代码存在变更时,创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务;其中,所述版本构建任务用于构建的变更后的所述代码的目标应用程序的版本;

[0136] 生成模块403,将所述版本构建任务调度至容器集群中的节点中运行,并在所述节

点中生成用于版本构建的缓存容器；其中，所述缓存容器用于对版本构建过程中的中间数据进行缓存；

[0137] 构建模块404，在所述缓存容器中创建并启动用于版本构建的编译容器，并将所述版本构建参数传入至所述编译容器，以使所述编译容器基于所述版本构建参数和所述中间数据构建生成包含所述目标应用程序的版本的镜像。

[0138] 在本实施例中，在创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务之前，还包括：

[0139] 检查模块405(图4未示出)，检查与变更后的所述代码对应合入所述目标应用程序的代码版本分支是否为允许执行构建状态；

[0140] 若是，则进一步创建与所述目标应用程序对应的携带了所述版本构建参数的版本构建任务；否则，禁止版本构建任务创建。

[0141] 在本实施例中，所述版本构建参数包括用于生成所述缓存容器的第一容器镜像，所述生成模块403进一步：

[0142] 将所述第一容器镜像作为模板，在所述节点中生成所述第一容器镜像的第一容器实例，将所述第一容器实例作为用于版本构建的缓存容器。

[0143] 在本实施例中，所述版本构建参数还包括用于生成所述编译容器的第二容器镜像、用于编译目标应用程序的代码的编译环境和编译脚本；

[0144] 所述构建模块404进一步：

[0145] 将所述第二容器镜像作为模板，在所述缓存容器中生成所述第二容器镜像的第二容器实例，将所述第二容器实例作为用于版本构建的编译容器；

[0146] 在所述编译容器启动运行后，将所述版本构建参数中的编译环境、编译脚本，作为参数传入至启动运行的所述编译容器。

[0147] 在本实施例中，所述版本构建参数还包括目标应用程序的代码获取地址，所述构建模块404进一步：

[0148] 所述编译容器从与基于所述代码获取地址对应的代码仓库中，获取目标应用程序的代码，并基于传入的所述版本构建参数中的编译环境、编译脚本，对所述代码执行版本编译，生成对应的目标应用程序的版本。

[0149] 在本实施例中，所述版本构建参数还包括作为目标应用程序的运行环境的第三容器镜像，所述构建模块404进一步：

[0150] 启动以所述第三容器镜像作为模板的第三容器实例；

[0151] 所述编译容器将生成的目标应用程序的版本，部署至所述第三容器实例中，并生成所述第三容器实例作为模板的镜像；

[0152] 并生成与所述应用镜像对应的应用镜像标签；其中，所述应用镜像标签至少包括目标应用程序对应的代码版本分支标签和目标应用程序的代码提交标签。

[0153] 在本实施例中，还包括：

[0154] 缓存模块406(图4未示出)，所述缓存容器将在所述编译容器生成目标应用程序的版本过程中所需的中间数据预先缓存在所述缓存容器本地的第一缓存目录下，以使所述编译容器基于所述缓存容器传入的环境变量，将所述第一缓存目录挂载到所述编译容器本地的第二缓存目录下，用于生成目标应用程序的版本。

[0155] 在本实施例中,所述中间数据至少包括容器镜像、版本编译依赖数据。

[0156] 在本实施例中,所述中间数据还包括版本编译过程的编译中间产物数据;在所述编译容器生成目标应用程序的版本过程中,所述缓存模块进一步:

[0157] 所述编译容器将所述编译中间产物数据保存至所述缓存容器中的第一缓存目录中进行缓存。

[0158] 在本实施例中,所述容器集群为基于Kubernetes管理的Docker容器集群。

[0159] 在本实施例中,所述缓存容器和所述编译容器,都为Docker容器。

[0160] 上述实施例阐明的装置、装置、模块或模块,具体可以由计算机芯片或实体实现,或者由具有某种功能的产品来实现。一种典型的实现设备为计算机,计算机的具体形式可以是个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件收发设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任意几种设备的组合。

[0161] 本领域技术人员在考虑说明书及实践这里公开的发明后,将容易想到本说明书的其它实施方案。本说明书旨在涵盖本说明书的任何变型、用途或者适应性变化,这些变型、用途或者适应性变化遵循本说明书的一般性原理并包括本说明书未公开的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的,本说明书的真正范围和精由下面的权利要求指出。

[0162] 应当理解的是,本说明书并不局限于上面已经描述并在附图中示出的精确结构,并且可以在不脱离其范围进行各种修改和改变。本说明书的范围仅由所附的权利要求来限制。

[0163] 以上所述仅为本说明书的较佳实施例而已,并不用以限制本说明书,凡在本说明书的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本说明书保护的范围内。

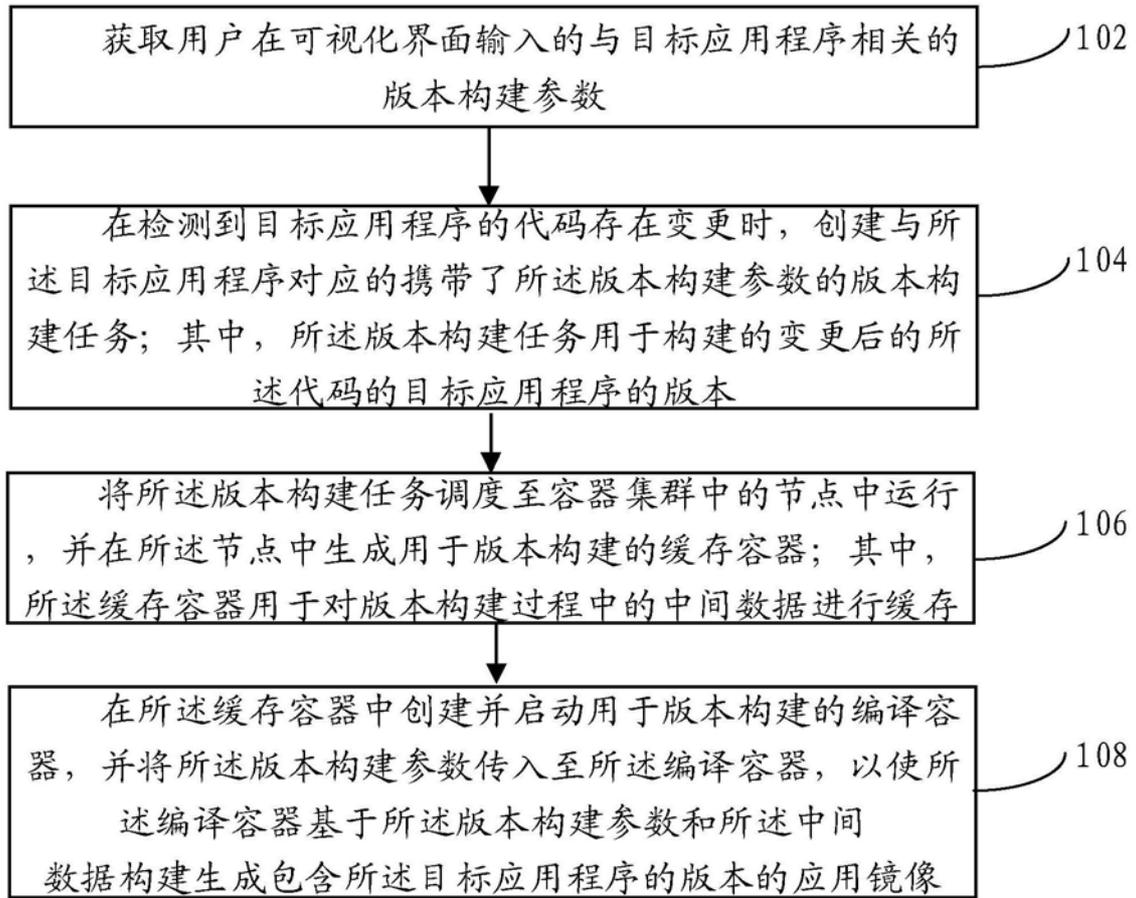


图1

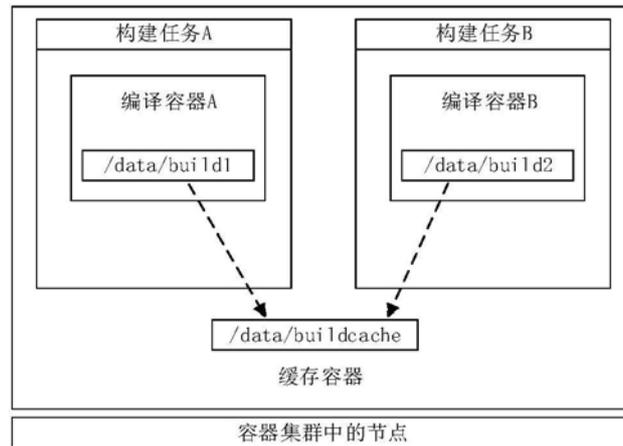


图2

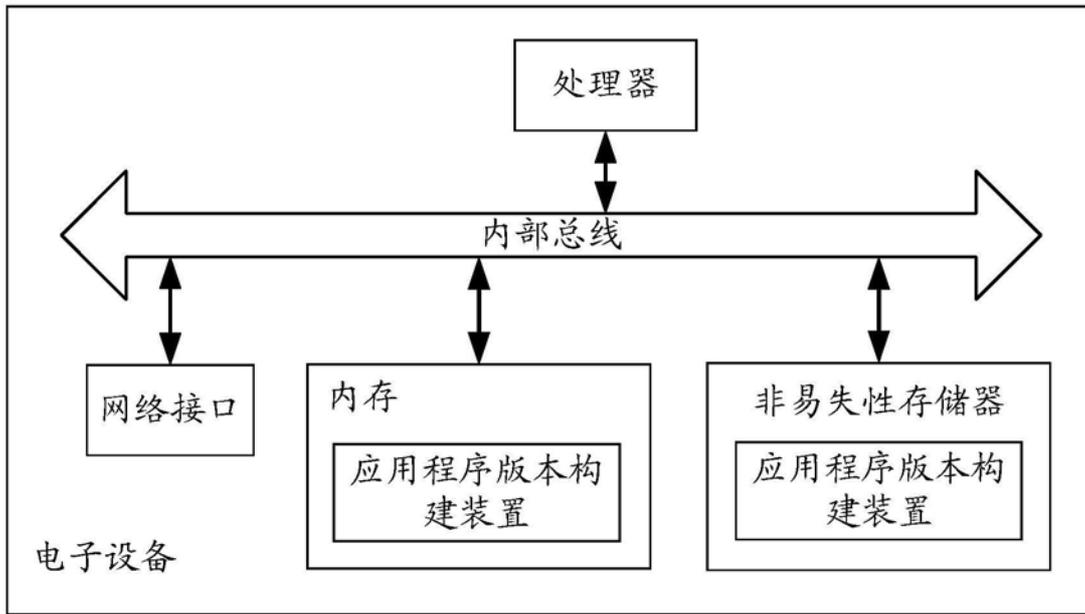


图3

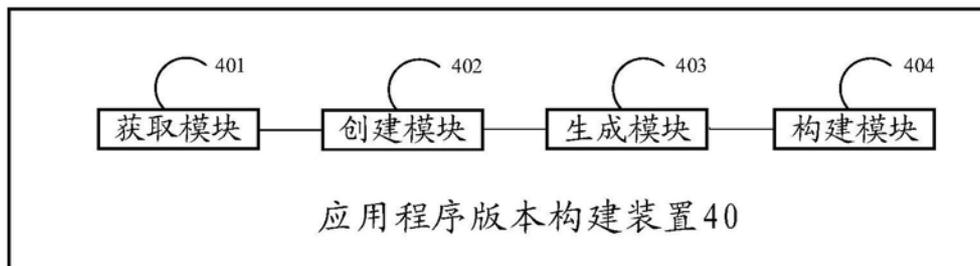


图4