(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 114489676 A
(43)申请公布日 2022.05.13

(21)申请号 202210135551.3

(22)申请日 2022.02.14

(71)申请人 中国工商银行股份有限公司
地址 100140 北京市西城区复兴门内大街55号

(72)发明人 雷经纬 徐嘉禛 翁晓俊 熊辉

(74)专利代理机构 北京康信知识产权代理有限责任公司 11240
专利代理师 黄海英

(51)Int.Cl.
G06F 8/41(2018.01)

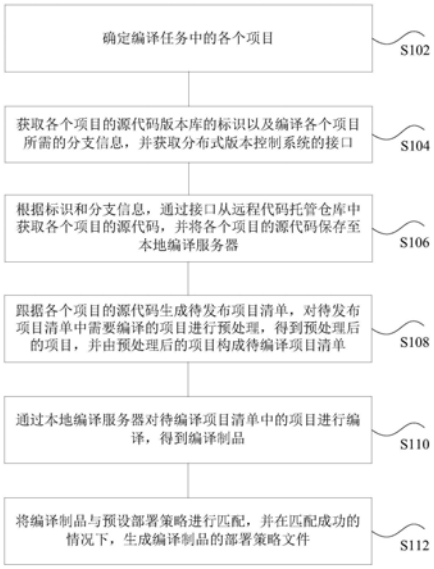
权利要求书3页 说明书14页 附图8页

(54)发明名称

源代码的处理方法、系统、存储介质及电子设备

(57)摘要

本申请公开了一种源代码的处理方法、系统、存储介质及电子设备。涉及金融科技领域，该方法包括：确定编译任务中的各个项目；获取各个项目的源代码版本库的标识以及编译各个项目所需的分支信息，并获取分布式版本控制系统的接口；获取各个项目的源代码，并将各个项目的源代码保存至本地编译服务器；依据各个项目的源代码生成待发布项目清单，对待发布项目清单中需要编译的项目进行预处理，并由预处理后的项目构成待编译项目清单；对待编译项目清单中的项目进行编译，得到编译制品；将编译制品与预设部署策略进行匹配，并在匹配成功的情况下，生成编译制品的部署策略文件。通过本申请，解决了相关技术中系统项目构建和部署效率低的问题。



CN 114489676 A

1.一种源代码的处理方法,其特征在于,包括:

确定编译任务中的各个项目;

获取所述各个项目的源代码版本库的标识以及编译所述各个项目所需的分支信息,并获取分布式版本控制系统的接口;

根据所述标识和所述分支信息,通过所述接口从远程代码托管仓库中获取所述各个项目的源代码,并将所述各个项目的源代码保存至本地编译服务器;

依据所述各个项目的源代码生成待发布项目清单,对所述待发布项目清单中需要编译的项目进行预处理,得到预处理后的项目,并由所述预处理后的项目构成待编译项目清单;

通过所述本地编译服务器对所述待编译项目清单中的项目进行编译,得到编译制品;

将所述编译制品与预设部署策略进行匹配,并在匹配成功的情况下,生成所述编译制品的部署策略文件。

2.根据权利要求1所述的方法,其特征在于,在将所述各个项目的源代码保存至本地编译服务器之后,该方法还包括:

检测所述源代码是否符合源代码管理规则;

在所述源代码符合源代码管理规则的情况下,执行依据所述各个项目的源代码生成待发布项目清单的步骤;

在所述源代码不符合源代码管理规则的情况下,淘汰所述源代码。

3.根据权利要求1所述的方法,其特征在于,根据所述标识和所述分支信息,通过所述接口从远程代码托管仓库中获取所述各个项目的源代码包括:

判断所述本地编译服务器是否存在每个项目的分支信息对应的源代码;

在所述本地编译服务器不存在所述分支信息对应的源代码的情况下,从所述远程代码托管仓库中获取所述分支信息对应的所有源代码;

在所述本地编译服务器存在所述分支信息对应的源代码的情况下,从所述远程代码托管仓库中获取所述分支信息对应的增量源代码,其中,所述增量源代码是与所述本地编译服务器中已存储的源代码不同的源代码。

4.根据权利要求1所述的方法,其特征在于,在对所述待发布项目清单中需要编译的项目进行预处理之前,该方法还包括:

判断所述待发布项目清单中的项目是否关联有预设标志信息,其中,所述预设标志信息用于指示项目需要进行编译;

根据是否关联所述预设标志信息对所述待发布项目清单中的项目进行分类,得到所述需要编译的项目和不需要编译的项目,其中,所述需要编译的项目为关联有所述预设标志信息的项目,所述不需要编译的项目为未关联有所述预设标志信息的项目。

5.根据权利要求1所述的方法,其特征在于,对所述待发布项目清单中需要编译的项目进行预处理包括:

判断所述待编译项目清单中的每个项目与其他项目之间是否存在依赖关系,其中,所述其他项目为所述编译项目清单中项目;

在一个项目与其他项目之间不存在所述依赖关系的情况下,确定所述项目的项目信息并标记依赖关系标识;

在一个项目与其他项目之间存在所述依赖关系的情况下,确定所述项目对应的依赖项

目，确定所述项目的信息并标记依赖关系标识，确定所述依赖项目的项目信息并标记依赖关系标识。

6.根据权利要求1所述的方法，其特征在于，对所述待发布项目清单中需要编译的项目进行预处理还包括：

判断所述待编译项目清单中的每个项目是否为增量部署项目；

在一个项目不是所述增量部署项目的情况下，对所述项目添加全量部署标记；

在一个项目是所述增量部署项目的情况下，对所述项目添加增量部署标记。

7.根据权利要求5所述的方法，其特征在于，通过所述本地编译服务器对所述待编译项目清单中的项目进行编译，得到编译制品包括：

为所述待编译项目清单中的每个项目分别分配一个线程；

在启动待编译的当前项目对应的线程之前，通过所述依赖关系标识判断所述待编译项目是否存在依赖项目；

在所述当前项目存在依赖项目的情况下，判断所述依赖项目是否为所述当前项目的父项目；

在所述依赖项目为所述父项目的情况下，停止执行所述当前项目的线程，并执行所述父项目对应的线程；

在所述依赖项目不是所述父项目的情况下，对所述当前项目进行编译，得到所述当前项目对应的编译制品。

8.根据权利要求1所述的方法，其特征在于，将所述编译制品与预设部署策略进行匹配包括：

获取所述编译制品和所述预设部署策略；

分别校验所述编译制品和所述预设部署策略是否符合预设规则；

在所述编译制品和所述预设部署策略均符合所述预设规则的情况下，执行将所述编译制品与所述预设部署策略进行匹配的步骤；

在所述编译制品和所述预设部署策略不符合所述预设规则的情况下，禁止执行将所述编译制品与所述预设部署策略进行匹配的步骤。

9.根据权利要求1所述的方法，其特征在于，在匹配成功的情况下，生成所述编译制品的部署策略文件包括：

根据所述编译制品的运维阶段对所述预设部署策略进行匹配，得到各个运维阶段对应的部署策略；

依据所述编译制品的运维阶段以及所述各个运维阶段对应的部署策略生成所述部署策略文件。

10.一种源代码的处理系统，其特征在于，包括：

控制单元，用于控制自检单元、源代码管理单元、项目编译单元、部署策略管理单元依次工作；

所述自检单元，用于执行预校验操作，其中，所述预校验操作至少包括校验系统的配置参数以及校验所述系统的外部连接设备；

所述源代码管理单元，用于生成源代码管理任务，获取所述源代码管理任务中的各个项目的源代码版本库的标识以及编译所述各个项目所需的分支信息，根据所述标识以及所

述分支信息将所述各个项目的源代码从远程代码托管仓库中下载至本地编译服务器；

项目编译单元，用于根据所述源代码管理任务中的项目确定项目编译任务，依据所述项目编译任务中各个项目的源代码生成待发布项目清单，对所述待发布项目清单中需要编译的项目进行预处理，得到预处理后的项目，由所述预处理后的项目构成待编译项目清单，并通过所述本地编译服务器对所述待编译项目清单中的项目进行编译，得到编译制品；

部署策略管理单元，用于将所述编译制品与预设部署策略进行匹配，并在匹配成功的情况下，生成所述编译制品的部署策略文件。

11.一种计算机存储介质，其特征在于，所述计算机存储介质用于存储程序，其中，所述程序运行时控制所述计算机存储介质所在的设备执行权利要求1至9中任意一项所述的源代码的处理方法。

12.一种电子设备，其特征在于，包含处理器和存储器，所述存储器中存储有计算机可读指令，所述处理器用于运行所述计算机可读指令，其中，所述计算机可读指令运行时执行权利要求1至9中任意一项所述的源代码的处理方法。

源代码的处理方法、系统、存储介质及电子设备

技术领域

[0001] 本申请涉及金融科技领域，具体而言，涉及一种源代码的处理方法、系统、存储介质及电子设备。

背景技术

[0002] 现有的系统项目构建和部署工具，是根据项目需求，由特定的构建管理工具组成，其中，部分通用的构建管理工具包括Ant(Apache Ant)、Maven和Jenkins，Apache Ant是一款适用于Java项目的标准开源构建工具。Ant使用易于理解的XML(Extensive Markup Language，可扩展标示语言)语言作为配置格式，用来在项目构建的流程中对具体参数进行配置；Maven是一个软件(尤其是基于Java开发的软件项目)项目管理和自动构建工具。Maven同样使用了XML作为配置格式，但它同时应用了统一的基于对象模型(POM)概念，通过POM可以达成一个中央信息片段能管理一个项目的构建、报告和文档等步骤的作用。Jenkins是一个开源的持续集成工具，可用于软件开发中的各种自动化任务，包括构建、测试和部署等，是系统项目构建和部署工具链的重要组成部分。但是Ant、Maven和Jenkins只能根据项目需求单独完成项目的部署或者构建任务，而不能完成项目的构建和部署任务。

[0003] 高质量、短周期的软件版本迭代对于系统项目构建和部署流程高度自动化的需求越来越紧迫，对于一个复杂系统中保护多个相互关联的大型Java项目而言，通过Java项目离线手动编译、程序安装包单点手动部署的方式，无法满足当今系统项目构建和部署对构建工具的效率、质量和自动化的要求。

[0004] 针对相关技术中系统项目构建和部署效率低的问题，目前尚未提出有效的解决方案。

发明内容

[0005] 本申请提供一种源代码的处理方法、系统、存储介质及电子设备，以解决相关技术中系统项目构建和部署效率低的问题。

[0006] 根据本申请的一个方面，提供了一种源代码的处理方法。该方法包括：确定编译任务中的各个项目；获取各个项目的源代码版本库的标识以及编译各个项目所需的分支信息，并获取分布式版本控制系统的接口；根据标识和分支信息，通过接口从远程代码托管仓库中获取各个项目的源代码，并将各个项目的源代码保存至本地编译服务器；依据各个项目的源代码生成待发布项目清单，对待发布项目清单中需要编译的项目进行预处理，得到预处理后的项目，并由预处理后的项目构成待编译项目清单；通过本地编译服务器对待编译项目清单中的项目进行编译，得到编译制品；将编译制品与预设部署策略进行匹配，并在匹配成功的情况下，生成编译制品的部署策略文件。

[0007] 可选地，在将各个项目的源代码保存至本地编译服务器之后，该方法还包括：检测源代码是否符合源代码管理规则；在源代码符合源代码管理规则的情况下，执行依据各个项目的源代码生成待发布项目清单的步骤；在源代码不符合源代码管理规则的情况下，淘

汰源代码。通过对源代码进行检测，保证在源代码处理后可以适配对应的系统。

[0008] 可选地，根据标识和分支信息，通过接口从远程代码托管仓库中获取各个项目的源代码包括：判断本地编译服务器是否存在每个项目的分支信息对应的源代码；在本地编译服务器不存在分支信息对应的源代码的情况下，从远程代码托管仓库中获取分支信息对应的所有源代码；在本地编译服务器存在分支信息对应的源代码的情况下，从远程代码托管仓库中获取分支信息对应的增量源代码，其中，增量源代码是与本地编译服务器中已存储的源代码不同的源代码。通过增量下载和全量下载两张方式获得源代码可以使本地编译服务器高效利用，不浪费下载源代码所需的带宽资源。

[0009] 可选地，在对待发布项目清单中需要编译的项目进行预处理之前，该方法还包括：判断待发布项目清单中的项目是否关联有预设标志信息，其中，预设标识信息用于指示项目需要进行编译；根据是否关联预设标志信息对待发布项目清单中的项目进行分类，得到需要编译的项目和不需要编译的项目，其中，需要编译的项目为关联有预设标志信息的项目，不需要编译的项目为未关联有预设标志信息的项目。通过区分是否需要编译的项目，可以让本地编译服务器在编译时不会重复编译造成资源浪费。

[0010] 可选地，对待发布项目清单中需要编译的项目进行预处理包括：判断待编译项目清单中的每个项目与其他项目之间是否存在依赖关系，其中，其他项目为编译项目清单中项目；在一个项目与其他项目之间不存在依赖关系的情况下，确定项目的项目信息并标记依赖关系标识；在一个项目与其他项目之间存在依赖关系的情况下，确定项目对应的依赖项目，确定项目的信息并标记依赖关系标识，确定依赖项目的项目信息并标记依赖关系标识。通过确认项目间的依赖关系，避免在编译时对有依赖关系的项目编译顺序混乱导致编译出错。

[0011] 可选地，对待发布项目清单中需要编译的项目进行预处理还包括：判断待编译项目清单中的每个项目是否为增量部署项目；在一个项目不是增量部署项目的情况下，对项目添加全量部署标记；在一个项目是增量部署项目的情况下，对项目添加增量部署标记。通过添加增量部署标记保证编译时对待编译项目区分增量编译还是全量编译。

[0012] 可选地，通过本地编译服务器对待编译项目清单中的项目进行编译，得到编译制品包括：为待编译项目清单中的每个项目分别分配一个线程；在启动待编译的当前项目对应的线程之前，通过依赖关系标识判断待编译项目是否存在依赖项目；在当前项目存在依赖项目的情况下，判断依赖项目是否为当前项目的父项目；在依赖项目为父项目的情况下，停止执行当前项目的线程，并执行父项目对应的线程；在依赖项目不是父项目的情况下，对当前项目进行编译，得到当前项目对应的编译制品。通过调整具有依赖关系的编译项目的编译顺序，可以让有依赖关系的项目按照先后顺序编译，不会造成编译错误。

[0013] 可选地，将编译制品与预设部署策略进行匹配包括：获取编译制品和预设部署策略；分别校验编译制品和预设部署策略是否符合预设规则；在编译制品和预设部署策略均符合预设规则的情况下，执行将编译制品与预设部署策略进行匹配的步骤；在编译制品和预设部署策略不符合预设规则的情况下，禁止执行将编译制品与预设部署策略进行匹配的步骤。通过将编译制品与预设部署策略匹配来判断本次源代码的处理结果是成功还是失败。

[0014] 可选地，在匹配成功的情况下，生成编译制品的部署策略文件包括：根据编译制品

的运维阶段对预设部署策略进行匹配，得到各个运维阶段对应的部署策略；依据编译制品的运维阶段以及各个运维阶段对应的部署策略生成部署策略文件。通过生成部署策略文件可以指导运维阶段对应系统模块的应用。

[0015] 根据本申请的另一方面，提供了一种源代码的处理系统。该系统包括：控制单元，用于控制自检单元、源代码管理单元、项目编译单元、部署策略管理单元依次工作；自检单元，用于执行预校验操作，其中，预校验操作至少包括校验系统的配置参数以及校验系统的外部连接设备；源代码管理单元，用于生成源代码管理任务，获取源代码管理任务中的各个项目的源代码版本库的标识以及编译各个项目所需的分支信息，根据标识以及分支信息将各个项目的源代码从远程代码托管仓库中下载至本地编译服务器；项目编译单元，用于根据源代码管理任务中的项目确定项目编译任务，依据项目编译任务中各个项目的源代码生成待发布项目清单，对待发布项目清单中需要编译的项目进行预处理，得到预处理后的项目，由预处理后的项目构成待编译项目清单，并通过本地编译服务器对待编译项目清单中的项目进行编译，得到编译制品；部署策略管理单元，用于将编译制品与预设部署策略进行匹配，并在匹配成功的情况下，生成编译制品的部署策略文件。

[0016] 根据本发明实施例的另一方面，还提供了一种计算机存储介质，计算机存储介质用于存储程序，其中，程序运行时控制非易失性存储介质所在的设备执行一种源代码的处理方法。

[0017] 根据本发明实施例的另一方面，还提供了一种电子设备，包含处理器和存储器；存储器中存储有计算机可读指令，处理器用于运行计算机可读指令，其中，计算机可读指令运行时执行一种源代码的处理方法。

[0018] 通过本申请，采用以下步骤：确定编译任务中的各个项目；获取各个项目的源代码版本库的标识以及编译各个项目所需的分支信息，并获取分布式版本控制系统的接口；根据标识和分支信息，通过接口从远程代码托管仓库中获取各个项目的源代码，并将各个项目的源代码保存至本地编译服务器；依据各个项目的源代码生成待发布项目清单，对待发布项目清单中需要编译的项目进行预处理，得到预处理后的项目，并由预处理后的项目构成待编译项目清单；通过本地编译服务器对待编译项目清单中的项目进行编译，得到编译制品；将编译制品与预设部署策略进行匹配，并在匹配成功的情况下，生成编译制品的部署策略文件，解决了相关技术中系统项目构建和部署效率低的问题。通过集成源码管理、项目编译和部署策略管理的功能统一对系统项目进行处理，进而达到了系统项目高效构建和部署的效果。

附图说明

[0019] 构成本申请的一部分的附图用来提供对本申请的进一步理解，本申请的示意性实施例及其说明用于解释本申请，并不构成对本申请的不当限定。在附图中：

[0020] 图1是根据本申请实施例提供的源代码的处理方法的流程图；

[0021] 图2是根据本申请实施例提供的本地编译服务器下载源代码的流程图；

[0022] 图3是根据本申请实施例提供的源代码对应项目部署策略管理的流程图；

[0023] 图4是根据本申请实施例提供的源代码的处理系统结构示意图；

[0024] 图5是根据本申请实施例提供的源代码的处理系统的外部连接环境示意图；

[0025] 图6是根据本申请实施例提供的项目编译的方法的流程图；

[0026] 图7是根据本申请实施例提供的源代码的处理装置的示意图；

[0027] 图8是根据本申请实施例提供的一种电子设备的示意图。

具体实施方式

[0028] 需要说明的是，在不冲突的情况下，本申请中的实施例及实施例中的特征可以相互组合。下面将参考附图并结合实施例来详细说明本申请。

[0029] 为了使本技术领域的人员更好地理解本申请方案，下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分的实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都应当属于本申请保护的范围。

[0030] 需要说明的是，本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换，以便这里描述的本申请的实施例。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0031] 为了便于描述，以下对本申请实施例涉及的部分名词或术语进行说明：

[0032] Maven:Apache Maven，一般简称为Maven，是一个软件(特别是Java软件)项目管理和自动构建工具，基于对象模型(POM)概念，利用一个中央信息片段能管理一个项目的构建、报告和文档等步骤。

[0033] 持续集成：是一种软件工程流程，将所有软件工程师对于软件的工作副本持续集成到共享主线mainline的一种举措，用于解决软件进行系统集成时面临的各项问题。

[0034] 编译：是一种利用编译程序从源语言的编写的源代码产生为目标程序的过程，可将高级语言编程计算机能识别的2进制语言。

[0035] Sql:Structured Query Language，是具有数据操作和数据定义等多种功能的数据库语言。

[0036] 根据本申请的实施例，提供了一种源代码的处理方法。

[0037] 图1是根据本申请实施例提供的源代码的处理方法的流程图。如图1所示，该方法包括以下步骤：

[0038] 步骤S102，确定编译任务中的各个项目。

[0039] 具体地，编译任务中的各个项目是指待更新源码的系统项目。

[0040] 步骤S104，获取各个项目的源代码版本库的标识以及编译各个项目所需的分支信息，并获取分布式版本控制系统的接口。

[0041] 具体地，源代码版本库是储存系统项目源代码的远程代码托管仓库，每个项目在源代码版本库中有对应的唯一标识，分支信息是指项目中包含的需要修改的源代码对应的信息，分布式版本控制系统的接口是源代码版本库的下载路径对应的接口。

[0042] 步骤S106，根据标识和分支信息，通过接口从远程代码托管仓库中获取各个项目

的源代码，并将各个项目的源代码保存至本地编译服务器。

[0043] 具体地，标识和分支信息可以确认需要更新的源代码所在的版本库和项目分支，通过接口可以将需要更新的源代码对应下载到本地的编译服务器。

[0044] 步骤S108，依据各个项目的源代码生成待发布项目清单，对待发布项目清单中需要编译的项目进行预处理，得到预处理后的项目，并由预处理后的项目构成待编译项目清单。

[0045] 具体地，待发布项目清单是系统项目对应的需要更新的项目清单，待发布项目清单中包含需要进行编译的项目和不需要编译的项目，不需要编译的项目不用做预处理，预处理是对需要编译的项目在编译前进行添加依赖关系标记以及增量更新标记，添加标记后可以对应的进行项目编译。

[0046] 步骤S110，通过本地编译服务器对待编译项目清单中的项目进行编译，得到编译制品。

[0047] 具体地，对待编译项目清单中的项目进行编译就是将待更新的源代码的配置进行更新，获得更新后的源代码，也即编译制品。

[0048] 步骤S112，将编译制品与预设部署策略进行匹配，并在匹配成功的情况下，生成编译制品的部署策略文件。

[0049] 具体地，预设部署策略是根据源代码的处理系统运维时涉及到的各个模块对应的项目进行部署的，匹配成功就是更新后的源代码对应的项目能在运维时正常使用。

[0050] 本申请实施例提供的源代码的处理方法，通过确定编译任务中的各个项目；获取各个项目的源代码版本库的标识以及编译各个项目所需的分支信息，并获取分布式版本控制系统的接口；根据标识和分支信息，通过接口从远程代码托管仓库中获取各个项目的源代码，并将各个项目的源代码保存至本地编译服务器；依据各个项目的源代码生成待发布项目清单，对待发布项目清单中需要编译的项目进行预处理，得到预处理后的项目，并由预处理后的项目构成待编译项目清单；通过本地编译服务器对待编译项目清单中的项目进行编译，得到编译制品；将编译制品与预设部署策略进行匹配，并在匹配成功的情况下，生成编译制品的部署策略文件，解决了相关技术中系统项目构建和部署效率低的问题。通过集成源码管理、项目编译和部署策略管理的功能统一对系统项目进行处理，进而达到了系统项目高效构建和部署的效果。

[0051] 由于在源代码版本库下载的源代码可能出现与本地编译服务器不适配的情况，可选地，在本申请实施例提供的源代码的处理方法中，在将各个项目的源代码保存至本地编译服务器之后，该方法还包括：检测源代码是否符合源代码管理规则；在源代码符合源代码管理规则的情况下，执行依据各个项目的源代码生成待发布项目清单的步骤；在源代码不符合源代码管理规则的情况下，淘汰源代码。

[0052] 具体地，源代码管理规则是判断下载的代码是否与本地编译服务器适配而设定的，例如配置代码评审规则、代码质量风险规则就属于源代码管理规则，如果下载的代码不能与本地编译服务器适配就需要重新下载可以适配的源代码。通过对源代码进行检测，保证在源代码处理后可以适配对应的系统。

[0053] 由于本地编译服务器中存在待更新项目的部分源代码，可选地，在本申请实施例提供的源代码的处理方法中，根据标识和分支信息，通过接口从远程代码托管仓库中获取

各个项目的源代码包括:判断本地编译服务器是否存在每个项目的分支信息对应的源代码;在本地编译服务器不存在分支信息对应的源代码的情况下,从远程代码托管仓库中获取分支信息对应的所有源代码;在本地编译服务器存在分支信息对应的源代码的情况下,从远程代码托管仓库中获取分支信息对应的增量源代码,其中,增量源代码是与本地编译服务器中已存储的源代码不同的源代码。

[0054] 具体地,图2是根据本申请实施例提供的本地编译服务器下载源代码的流程图,如图2所示,判断本地编译服务器中源代码仓库的状态,检查是否存在每个项目的各个分支信息对应的源代码,例如其中一个项目的分支信息包括A分支信息和B分支信息,如果本地编译服务器中源代码仓库没有A分支信息对应的源代码,则通过下载路径下载时,对A分支对应的源代码进行全量下载(即clone操作),也即将远端服务器中A分支对应的所有源代码下载至本地编译服务器;如果本地编译服务器中源代码仓库有B分支信息对应的源代码,则通过下载路径下载时,对B分支对应的源代码进行增量下载(即pull操作),也即仅更新、合并远端服务器与本地编译服务器源代码差异的部分。在进行下载时判断源代码更新进度,向源代码的处理系统返回更新进度的状态码。通过增量下载和全量下载两张方式获得源代码可以使本地编译服务器高效利用,不浪费下载源代码所需的带宽资源。

[0055] 由于下载的源代码中还包括部分需要更新的源代码,可选地,在本申请实施例提供的源代码的处理方法中,在对待发布项目清单中需要编译的项目进行预处理之前,该方法还包括:判断待发布项目清单中的项目是否关联有预设标志信息,其中,预设标识信息用于指示项目需要进行编译;根据是否关联预设标志信息对待发布项目清单中的项目进行分类,得到需要编译的项目和不需要编译的项目,其中,需要编译的项目为关联有预设标志信息的项目,不需要编译的项目为未关联有预设标志信息的项目。

[0056] 具体地,预设标志信息是对象模型配置文件(即pom.xml)对应的标志位,通过对获取的待发布项目清单做逐行解析,根据是否需要编译,将清单中的项目分为两类,判断的方法为递归扫描清单中的项目以及对应的父目录和同级目录,以是否包含Maven必备的项目对象模型配置文件作为需要编译的标志位,不需要编译的项目分支没有对象模型配置文件标志位,需要编译的项目分支有对象模型配置文件标志位。对于不需要编译的分支:例如配置文件、与配置相关的数据库sql脚本等,此类分支直接转移至编译制品;对于需要编译的分支,这些分支通常是java项目中包含的文件,需要通过编译器完成编译后才能在服务器上运行,此类分支由本地编译服务器进行编译。通过区分是否需要编译的项目,可以让本地编译服务器在编译时不会重复编译造成资源浪费。

[0057] 由于待编译的项目各分支间可能存在依赖关系,可选地,在本申请实施例提供的源代码的处理方法中,对待发布项目清单中需要编译的项目进行预处理包括:判断待编译项目清单中的每个项目与其他项目之间是否存在依赖关系,其中,其他项目为编译项目清单中项目;在一个项目与其他项目之间不存在依赖关系的情况下,确定项目的项目信息并标记依赖关系标识;在一个项目与其他项目之间存在依赖关系的情况下,确定项目对应的依赖项目,确定项目的信息并标记依赖关系标识,确定依赖项目的项目信息并标记依赖关系标识。

[0058] 具体地,逐条处理待发布清单中需要编译的分支,需要编译的分支包含对象模型配置文件,通过解析项目对象模型配置文件,判断待发布清单中不同分支之间是否存在依

赖关系。依赖关系是通过检查不同分支之间的配置是否会相互影响来判断的，例如A分支和B分支存在依赖关系，那么对A分支对应的参数进行更新后，B分支对应的参数也需要做出调整。如果待发布分支之间不存在依赖关系，收集待发布清单中待编译项目的信息并标记编译要素，其中编译要素包括项目对象模型配置、增量更新标记、依赖关系解析等；如果待发布分支之间存在依赖关系，根据获取的存在依赖关系的A、B分支，解析A、B所在Maven项目的项目对象模型配置文件，确定A、B分支的父子依赖关系，也就是确定更新A分支的参数会影响B分支，则A分支为父项目分支，对B分支添加父项目依赖标识，或者更新B分支的参数会影响A分支，则B分支为父项目分支，对A分支添加父项目依赖标识。通过确认项目间的依赖关系，避免在编译时对有依赖关系的项目编译顺序混乱导致编译出错。

[0059] 由于编译时会出现对增量下载的源代码和全量下载的源代码编译的情况，可选地，在本申请实施例提供的源代码的处理方法中，对待发布项目清单中需要编译的项目进行预处理还包括：判断待编译项目清单中的每个项目是否为增量部署项目；在一个项目不是增量部署项目的情况下，对项目添加全量部署标记；在一个项目是增量部署项目的情况下，对项目添加增量部署标记。

[0060] 具体地，根据待发布项目清单，逐条判断传入的待发布分支是否为增量部署项目。增量部署项目由于仅修改业务逻辑，所以项目改动量通常很小，实际编译和部署仅更新需要改动的部分即可生效，增量更新以“小步快跑”的节奏推进开发和测试，可有效提升版本迭代编译部署的效率。增量部署项目的判断依据为待发布分支对应的变更清单是否仅包含Java项目源码文件，例如C项目为增量部署项目，说明在本次编译任务中C项目只有业务逻辑的修改，没有新增的依赖包，是一个增量部署项目，对C项目进行增量编译；例如D项目不是增量部署项目，D项目对应的变更清单包含项目对象模型配置文件、配置文件等，说明在本次编译任务中D项目可能新增了依赖包或配置变量，需要对D项目进行全量编译。对需要全量部署的项目，添加全量部署标记，对增量部署项目，进一步解析对象模型配置文件，识别待编译项目的目标jar文件，对增量部署项目进行增量部署标记，完成待编译项目清单的解析，生成包含了编译要素标记的待编译项目清单，对待编译项目清单中的项目进行并行编译。通过添加增量部署标记保证编译时对待编译项目区分增量编译还是全量编译。

[0061] 在并行编译的过程中会碰到标记有依赖标识的项目，可选地，在本申请实施例提供的源代码的处理方法中，通过本地编译服务器对待编译项目清单中的项目进行编译，得到编译制品包括：为待编译项目清单中的每个项目分别分配一个线程；在启动待编译的当前项目对应的线程之前，通过依赖关系标识判断待编译项目是否存在依赖项目；在当前项目存在依赖项目的情况下，判断依赖项目是否为当前项目的父项目；在依赖项目为父项目的情况下，停止执行当前项目的线程，并执行父项目对应的线程；在依赖项目不是父项目的情况下，对当前项目进行编译，得到当前项目对应的编译制品。

[0062] 具体地，并行编译开始前，判断本线程的项目是否存在父项目依赖，例如E项目存在父项目依赖，也即E项目携带有父项目依赖标识，则线程进行优先级调整，E项目进入休眠和阻塞状态，在E项目的依赖项目编译完成后对E项目重新开始编译；例如F项目不存在父项目依赖标识，直接调用编译方法进行编译。对于编译失败的项目，线程允许失败项目三次重试编译，对于编译成功次数大于1的项目，将完成编译的状态信息发送给源代码的处理系统。编译成功后获得的编译制品按照预定的规则，例如是否为二进制文件、是否为配置文

件、是否与环境信息紧耦合等要素，对编译制品中的文件对象清单分门别类归置，将归置好的编译制品根据配置参数，按照一定命名规则，对编译制品进行压缩，向源代码的处理系统发送编译任务完成信号。通过调整具有依赖关系的编译项目的编译顺序，可以让有依赖关系的项目按照先后顺序编译，不会造成编译错误。

[0063] 编译完成后获得的编译制品可能与部署策略不适配，可选地，在本申请实施例提供的源代码的处理方法中，将编译制品与预设部署策略进行匹配包括：获取编译制品和预设部署策略；分别校验编译制品和预设部署策略是否符合预设规则；在编译制品和预设部署策略均符合预设规则的情况下，执行将编译制品与预设部署策略进行匹配的步骤；在编译制品和预设部署策略不符合预设规则的情况下，禁止执行将编译制品与预设部署策略进行匹配的步骤。

[0064] 具体地，部署策略是对编译制品分发给运维系统的规则，部署策略包括规则触发条件、被分发的服务器信息、执行分发的操作系统用户、被分发介质的权限、自定制执行操作系统脚本等。图3是根据本申请实施例提供的源代码对应项目部署策略管理的流程图，如图3所示，源代码的处理系统调起一个新部署策略管理任务，将编译制品的信息和预置的部署策略配置作为入口参数提交部署策略管理单元，部署策略管理单元将接收的参数及本地配置文件进行合法校验，然后加载预设部署策略，将获取的预设部署策略与编译制品进行匹配，如编译制品与预置部署策略配置不符，也就是编译制品不能完成预设部署策略分配的任务，则本次系统项目的构建无法完成，向源代码的处理系统发出报错信号；如编译制品与预置部署策略配置可匹配，也就是编译制品可以完成预设部署策略分配的任务，说明本次系统项目的构建完成。通过将编译制品与预设部署策略匹配来判断本次源代码的处理结果是成功还是失败。

[0065] 系统项目构建完成后，该系统项目进入运维阶段，可选地，在本申请实施例提供的源代码的处理方法中，在匹配成功的情况下，生成编译制品的部署策略文件包括：根据编译制品的运维阶段对预设部署策略进行匹配，得到各个运维阶段对应的部署策略；依据编译制品的运维阶段以及各个运维阶段对应的部署策略生成部署策略文件。

[0066] 具体地，如图3所示，从本地编译服务器的配置文件中读取各个系统功能模块的部署策略，与系统运维阶段的操作步骤紧密相关。每一个模块的部署策略由多个程序部署要素组成，根据系统项目构建与部署工具的运维阶段操作步骤的分类，可将部署策略按照优先级排序，分为目录预处理、更新介质传输、配置文件传输、功能启动、自定义脚本运行五个模块。根据对编译制品和部署策略的解析结果，生成一份可供运维阶段使用的格式化自动化部署策略文件，与本次构建任务产生的编译制品一一对应，可供系统项目构建与部署工具自动化运维使用。通过生成部署策略文件可以指导运维阶段对应系统模块的应用。

[0067] 根据本申请的另一实施例，提供了一种源代码的处理系统。图4是根据本申请实施例提供的源代码的处理系统结构示意图，如图4所示，该系统包括：控制单元401，用于控制自检单元402、源代码管理单元403、项目编译单元404、部署策略管理单元依次工作405；自检单元402，用于执行预校验操作，其中，预校验操作至少包括校验系统的配置参数以及校验系统的外部连接设备；源代码管理单元403，用于生成源代码管理任务，获取源代码管理任务中的各个项目的源代码版本库的标识以及编译各个项目所需的分支信息，根据标识以及分支信息将各个项目的源代码从远程代码托管仓库中下载至本地编译服务器；项目编译

单元404，用于根据源代码管理任务中的项目确定项目编译任务，依据项目编译任务中各个项目的源代码生成待发布项目清单，对待发布项目清单中需要编译的项目进行预处理，得到预处理后的项目，由预处理后的项目构成待编译项目清单，并通过本地编译服务器对待编译项目清单中的项目进行编译，得到编译制品；部署策略管理单元405，用于将编译制品与预设部署策略进行匹配，并在匹配成功的情况下，生成编译制品的部署策略文件。

[0068] 具体地，本发明提供一种源代码的处理系统，该系统包括：控制单元401，用于提供总体流程控制和失败任务重试等功能，由控制单元401所连接起来的自检单元402、源码管理单元403、项目编译单元404、部署策略管理单元405组成本系统的四个主要部分。自检单元402由参数解析模块21、环境检查模块22组成，对输入参数进行合法性校验、对必要配置参数进行加载、对环境可用性进行预检查，如前置校验流程通过、所有配置参数加载完成，则将信号发送给控制单元401，表示系统已进入运行态。

[0069] 源码管理单元403由版本控制系统对接模块31和代码库管理模块32组成，为源代码的处理系统提供了本地源代码仓库和源代码版本控制的功能，确保源代码对应的软件系统在经过不同特性开发和持续集成后，作为一个完整的整体对外发布。版本控制系统对接模块31可与成熟的分布式版本控制工具集成，分布式版本控制工具包括Git、Gitlab等。代码库管理模块32可根据项目需求为开发中的软件系统提供定制化的源代码管理规则，例如配置代码评审规则、代码质量风险规则等。

[0070] 项目编译单元404由增量程序清单识别模块41、Java项目构建配置解析模块42、编译输出物分类模块43组成，是源代码的处理系统的重要组成部分，增量程序清单识别模块41和Java项目构建配置解析模块42用于对待发布项目清单中需要编译的项目进行预处理，编译输出物分类模块43用于待编译项目清单中的项目进行编译，得到编译制品。

[0071] 部署策略管理单元405由部署规则解析模块51、部署策略生成模块52组成，是一个面向程序部署自动化的管理单元，为源代码对应的系统项目提供自动化部署的配置管理工具链。通过对软件所部署的基础设备作为“代码”进行配置化管理，开发态和运行态的系统共享同质化的基础环境，来降低由于环境差异导致的生产问题。

[0072] 需要说明的是，图5是根据本申请实施例提供的源代码的处理系统的外部连接环境示意图，如图5所示，系统项目构建和部署系统由分布式版本控制系统501、持续集成系统502、系统项目构建和部署工作流系统503组成，复杂IT系统每一次更新都需要开发人员从分布式版本控制系统501中发起变更，经过了持续集成系统502连接到系统项目构建和部署工作流系统503，完整经历一次系统项目构建和部署工作流的三个阶段才能完成发布。分布式版本控制系统501是项目的代码仓库，其下包含若干个代码版本库，需按照功能区分，分为存放参数或配置表的代码库和存放系统特性功能的代码库1-代码库n；系统项目构建和部署工作流系统503可分为源代码的处理系统、部署策略系统、运维系统三个阶段，本发明所提及的源代码的处理系统是系统项目构建和部署工作流系统开始的第一个处理流程，连接部署策略系统和运维系统。

[0073] 根据本申请提供的另一个实施例，提供了一种项目编译的方法。

[0074] 图6是根据本申请实施例提供的项目编译的方法的流程图，如图6所示，主控程序将调起一个新项目编译任务，并向编译进程传递所有必要参数和配置文件，如源码版本库分支信息、源代码仓库存放目录、待发布项目清单、编译制品目录等。接下里对主控程序传

入的参数和本地配置文件进行加载和解析，完成入口参数和本地配置文件解析后，联动源码管理单元的分布式版本控制产品，结合入口参数中的版本控制标签信息，获取与本次编译任务相关的待发布程序或配置清单。对获取的待发布项目清单做逐行解析，根据是否需要编译，将清单中的条目分为两类，判断的方法为递归扫描登记条目以及其父目录和同级目录，以是否包含Maven必备的项目对象模型配置文件(即pom.xml)作为需要编译的标志位。对于不需要编译的条目：例如配置文件、与配置相关的数据库sql脚本等，此类条目直接交由不需要编译分支的编译制品拉取模块，将配置的工作目录按照预定的规则，例如是否为二进制文件、是否为配置文件、是否与环境信息紧耦合等要素，对传入的文件对象清单分门别类归置。解析传入的编译制品所在目录，根据配置参数，按照一定命名规则，对编译制品进行压缩，向控制单元发送编译任务完成信号。

[0075] 对于需要编译的条目，通常为java项目中包含的文件，需要通过编译器完成编译后才能在服务器上运行，此类条目交由编译分支处理，通过对待发布清单进行逐条解析，输出一个由项目对象模型配置(即pom.xml)、增量更新标记、依赖关系解析等编译要素组成的清单对象，具体步骤如下：逐条处理待发布清单中需要编译的条目，解析项目对象模型配置文件(pom.xml)，判断待发布清单中若干条目之间是否存在依赖关系。如待发布条目之间不存在依赖关系，则收集待发布清单中待编译项目的信息并标记编译要素，并以对象追加的形式将待编译项目信息对象提交至识别编译目标jar，其中编译要素包括项目对象模型配置(即pom.xml)、增量更新标记、依赖关系解析等。由于待发布项目清单可能不包含编译父项目，因此最终的待编译项目数量大于或等于待发布项目数量。

[0076] 如待发布条目之间存在依赖关系，根据获取的存在依赖关系的待发布条目，解析其所在Maven项目的项目对象模型配置文件(pom.xml)，确定待发布项目的父子依赖关系，并通过异步的方式向父子项目对象信息发送至编译要素标记模块。根据获取的待发布清单，判断传入的待发布条目是否为增量部署项目。将是否需要增量部署作为判断依据，原因是仅修改业务逻辑的项目改动量通常很小，实际编译和部署仅更新改动的部分即可生效，增量更新以“小步快跑”的节奏推进开发和测试，可有效提升版本迭代编译部署的效率。增量部署判断依据如下：如待发布条目的变更清单仅包含Java项目源码文件，说明在本次编译任务中只有业务逻辑的修改，没有新增的依赖包，是一个增量部署项目，进一步解析对象模型配置文件(pom.xml)，识别编译项目的目标jar文件，对待编译项目进行增存量部署属性标记，并向预编译pom.xml列表清单发起对象属性异步更新请求；如待发布条目的变更清单包含项目对象模型配置文件(pom.xml)、配置文件(application.properties)等，说明本次编译任务可能新增了依赖包或配置变量，需要对项目进行全量编译。获取需要全量部署的项目，添加全量部署标记，并向预编译pom.xml列表清单发起对象属性异步更新请求。

[0077] 完成待编译项目清单的解析，生成包含了编译要素标记的待编译项目列表清单解析，将清单提交到编译模块中并行编译。接收传入的待编译项目清单，完成线程池初始化，为每一个待编译任务分配一个线程，进入并行编译。并行编译开始前，判断本线程的项目是否存在父项目依赖，如存在父项目依赖，则线程将优先级调整请求发送至并行编译流程中的调度模块，负责更新待编译项目清单的状态、决定分配的每一个持有编译任务的线程生命周期，具体而言，包括编译完成情况的标记，处理即将进入休眠或阻塞状态线程优先级调整请求。在所有待编译项目列表清单处理完成后，发起编译制品转移的请求，随后进入休眠

和阻塞状态;如不存在父项目依赖,则线程进入线程的编译完成情况计数模块,对于编译失败的项目,允许三次重试,重新提交到线程中;对于编译成功次数大于1的项目,将完成情况更新请求发送至并行编译流程中的调度模块;将编译制品所在目录信息提交到后续模块,解析传入的编译制品所在目录,根据配置参数,按照一定命名规则,对编译制品进行压缩,向控制单元发送编译任务完成信号。

[0078] 需要说明的是,在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中执行,并且,虽然在流程图中示出了逻辑顺序,但是在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤。

[0079] 本申请实施例还提供了一种源代码的处理装置,需要说明的是,本申请实施例的源代码的处理装置可以用于执行本申请实施例所提供的用于源代码的处理方法。以下对本申请实施例提供的源代码的处理装置进行介绍。

[0080] 图7是根据本申请实施例的源代码的处理装置的示意图。如图7所示,该装置包括:确定单元10,用于确定编译任务中的各个项目;获取单元20,用于获取各个项目的源代码版本库的标识以及编译各个项目所需的分支信息,并获取分布式版本控制系统的接口;存储单元30,用于根据标识和分支信息,通过接口从远程代码托管仓库中获取各个项目的源代码,并将各个项目的源代码保存至本地编译服务器;生成单元40,用于依据各个项目的源代码生成待发布项目清单,对待发布项目清单中需要编译的项目进行预处理,得到预处理后的项目,并由预处理后的项目构成待编译项目清单;编译单元50,用于通过本地编译服务器对待编译项目清单中的项目进行编译,得到编译制品;匹配单元60,用于将编译制品与预设部署策略进行匹配,并在匹配成功的情况下,生成编译制品的部署策略文件。

[0081] 本申请实施例提供的源代码的处理装置,通过确定单元10,用于确定编译任务中的各个项目;获取单元20,用于获取各个项目的源代码版本库的标识以及编译各个项目所需的分支信息,并获取分布式版本控制系统的接口;存储单元30,用于根据标识和分支信息,通过接口从远程代码托管仓库中获取各个项目的源代码,并将各个项目的源代码保存至本地编译服务器;生成单元40,用于依据各个项目的源代码生成待发布项目清单,对待发布项目清单中需要编译的项目进行预处理,得到预处理后的项目,并由预处理后的项目构成待编译项目清单;编译单元50,用于通过本地编译服务器对待编译项目清单中的项目进行编译,得到编译制品;匹配单元60,用于将编译制品与预设部署策略进行匹配,并在匹配成功的情况下,生成编译制品的部署策略文件,解决了相关技术中系统项目构建和部署效率低的问题,通过集成源码管理、项目编译和部署策略管理的功能统一对系统项目进行处理,进而达到了系统项目高效构建和部署的效果。

[0082] 可选地,在本申请实施例提供的源代码的处理装置中,该装置还包括:检测单元,用于检测源代码是否符合源代码管理规则;在源代码符合源代码管理规则的情况下,执行依据各个项目的源代码生成待发布项目清单的步骤;在源代码不符合源代码管理规则的情况下,淘汰源代码。

[0083] 可选地,在本申请实施例提供的源代码的处理装置中,存储单元30包括:第一判断模块,用于判断本地编译服务器是否存在每个项目的分支信息对应的源代码;第一获取模块,用于在本地编译服务器不存在分支信息对应的源代码的情况下,从远程代码托管仓库中获取分支信息对应的所有源代码;第二获取模块,用于在本地编译服务器存在分支信息

对应的源代码的情况下，从远程代码托管仓库中获取分支信息对应的增量源代码，其中，增量源代码是与本地编译服务器中已存储的源代码不同的源代码。

[0084] 可选地，在本申请实施例提供的源代码的处理装置中，生成单元40包括：第二判断模块，用于判断待发布项目清单中的项目是否关联有预设标志信息，其中，预设标识信息用于指示项目需要进行编译；分类模块，用于根据是否关联预设标志信息对待发布项目清单中的项目进行分类，得到需要编译的项目和不需要编译的项目，其中，需要编译的项目为关联有预设标志信息的项目，不需要编译的项目为未关联有预设标志信息的项目。

[0085] 可选地，在本申请实施例提供的源代码的处理装置中，生成单元40还包括：第三判断模块，用于判断待编译项目清单中的每个项目与其他项目之间是否存在依赖关系，其中，其他项目为编译项目清单中项目；第一确定模块，用于在一个项目与其他项目之间不存在依赖关系的情况下，确定项目的项目信息并标记依赖关系标识；第二确定模块，用于在一个项目与其他项目之间存在依赖关系的情况下，确定项目对应的依赖项目，确定项目的信息并标记依赖关系标识，确定依赖项目的项目信息并标记依赖关系标识。

[0086] 可选地，在本申请实施例提供的源代码的处理装置中，生成单元40还包括：第四判断模块，用于判断待编译项目清单中的每个项目是否为增量部署项目；第一标记模块，用于在一个项目不是增量部署项目的情况下，对项目添加全量部署标记；第二标记模块，用于在一个项目是增量部署项目的情况下，对项目添加增量部署标记。

[0087] 可选地，在本申请实施例提供的源代码的处理装置中，编译单元50包括：分配模块，用于为待编译项目清单中的每个项目分别分配一个线程；第五判断模块，用于在启动待编译的当前项目对应的线程之前，通过依赖关系标识判断待编译项目是否存在依赖项目；第五判断模块包括判断子模块，用于在当前项目存在依赖项目的情况下，判断依赖项目是否为当前项目的父项目；第一执行模块，用于在依赖项目为父项目的情况下，停止执行当前项目的线程，并执行父项目对应的线程；编译模块，用于在依赖项目不是父项目的情况下，对当前项目进行编译，得到当前项目对应的编译制品。

[0088] 可选地，在本申请实施例提供的源代码的处理装置中，匹配单元60包括：第三获取模块，用于获取编译制品和预设部署策略；校验模块，用于分别校验编译制品和预设部署策略是否符合预设规则；第二执行模块，用于在编译制品和预设部署策略均符合预设规则的情况下，执行将编译制品与预设部署策略进行匹配的步骤；禁止模块，用于在编译制品和预设部署策略不符合预设规则的情况下，禁止执行将编译制品与预设部署策略进行匹配的步骤。

[0089] 可选地，在本申请实施例提供的源代码的处理装置中，匹配单元60还包括：匹配模块，用于根据编译制品的运维阶段对预设部署策略进行匹配，得到各个运维阶段对应的部署策略；生成模块，用于依据编译制品的运维阶段以及各个运维阶段对应的部署策略生成部署策略文件。

[0090] 上述源代码的处理装置包括处理器和存储器，上述确定单元10、获取单元20、存储单元30、生成单元40、编译单元50和匹配单元60等均作为程序单元存储在存储器中，由处理器执行存储在存储器中的上述程序单元来实现相应的功能。

[0091] 处理器中包含内核，由内核去存储器中调取相应的程序单元。内核可以设置一个或以上，通过调整内核参数来达到系统项目自动化高效构建和部署的效果。

[0092] 存储器可能包括计算机可读介质中的非永久性存储器,随机存取存储器(RAM)和/或非易失性内存等形式,如只读存储器(ROM)或闪存(flash RAM),存储器包括至少一个存储芯片。

[0093] 本申请实施例还提供了一种计算机存储介质,计算机存储介质用于存储程序,其中,程序运行时控制非易失性存储介质所在的设备执行一种源代码的处理方法。

[0094] 本申请实施例还提供了一种电子设备,图8是根据本申请实施例提供的一种电子设备的示意图,如图8所示,电子设备801包含处理器和存储器;存储器中存储有计算机可读指令,处理器用于运行计算机可读指令,其中,计算机可读指令运行时执行一种源代码的处理方法。本文中的电子设备可以是服务器、PC、PAD、手机等。

[0095] 本领域内的技术人员应明白,本申请的实施例可提供为方法、系统、或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0096] 本申请是参照根据本申请实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0097] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0098] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0099] 在一个典型的配置中,计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[0100] 存储器可能包括计算机可读介质中的非永久性存储器,随机存取存储器(RAM)和/或非易失性内存等形式,如只读存储器(ROM)或闪存(flash RAM)。存储器是计算机可读介质的示例。

[0101] 计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括,但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带,磁带磁盘存储或其他磁性存储设备或

任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括暂存电脑可读媒体(transitory media)，如调制的数据信号和载波。

[0102] 还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括要素的过程、方法、商品或者设备中还存在另外的相同要素。

[0103] 以上仅为本申请的实施例而已，并不用于限制本申请。对于本领域技术人员来说，本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等，均应包含在本申请的权利要求范围之内。

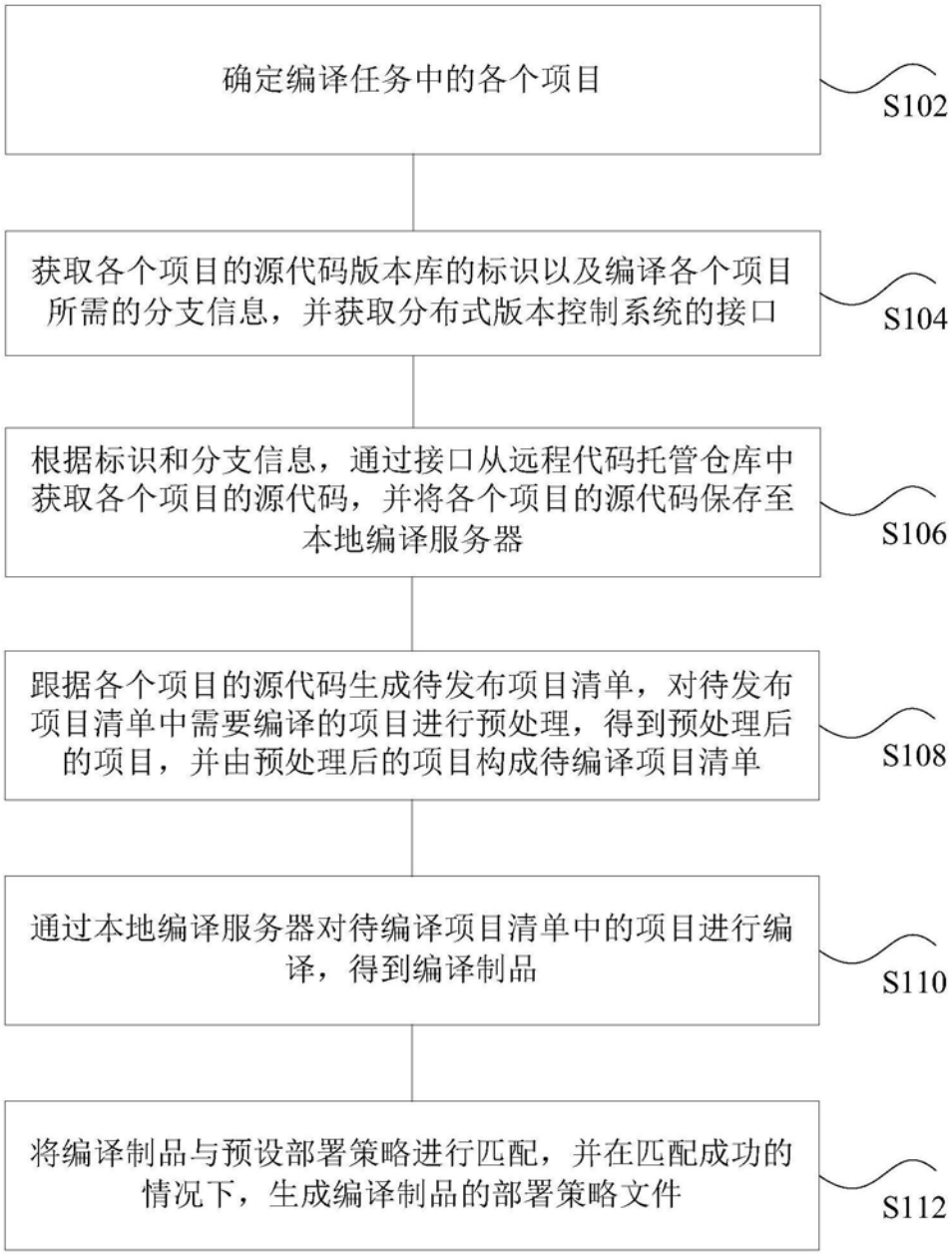

确定编译任务中的各个项目
S102
获取各个项目的源代码版本库的标识以及编译各个项目所需的分支信息，并获取分布式版本控制系统的接口
S104
根据标识和分支信息，通过接口从远程代码托管仓库中获取各个项目的源代码，并将各个项目的源代码保存至本地编译服务器
S106
跟据各个项目的源代码生成待发布项目清单，对待发布项目清单中需要编译的项目进行预处理，得到预处理后的项目，并由预处理后的项目构成待编译项目清单
S108
通过本地编译服务器对待编译项目清单中的项目进行编译，得到编译制品
S110
将编译制品与预设部署策略进行匹配，并在匹配成功的情况下，生成编译制品的部署策略文件
S112

图1

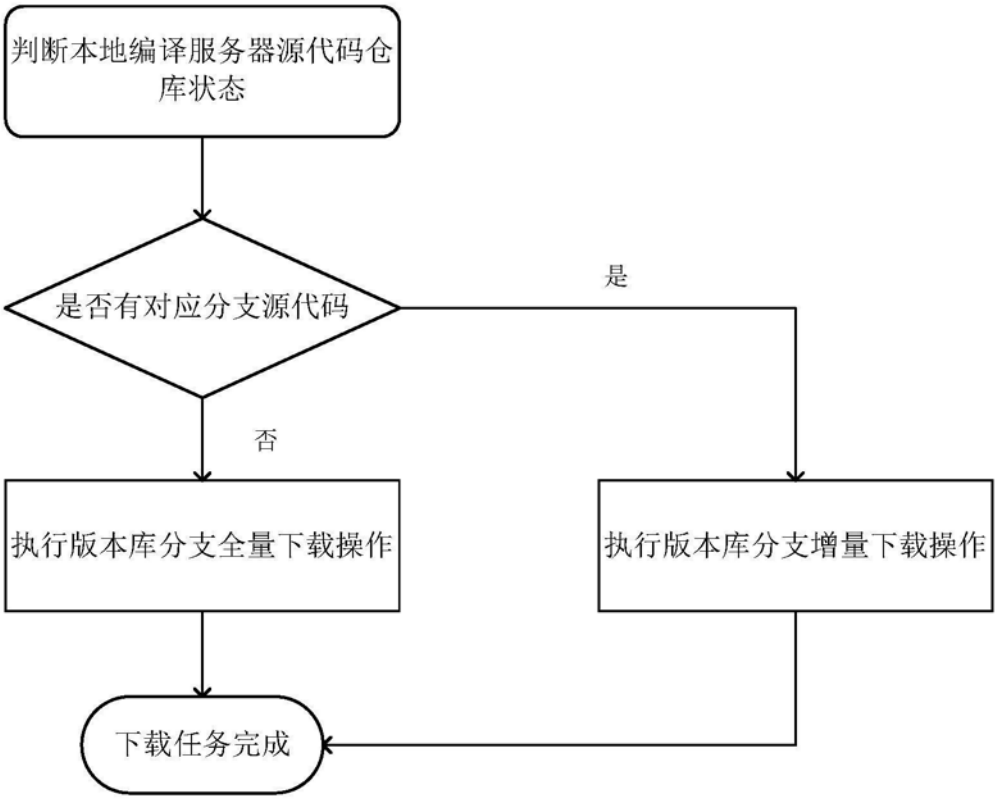

判断本地编译服务器源代码仓库状态
是否有对应分支源代码
是
否
执行版本库分支全量下载操作
执行版本库分支增量下载操作
下载任务完成

图2

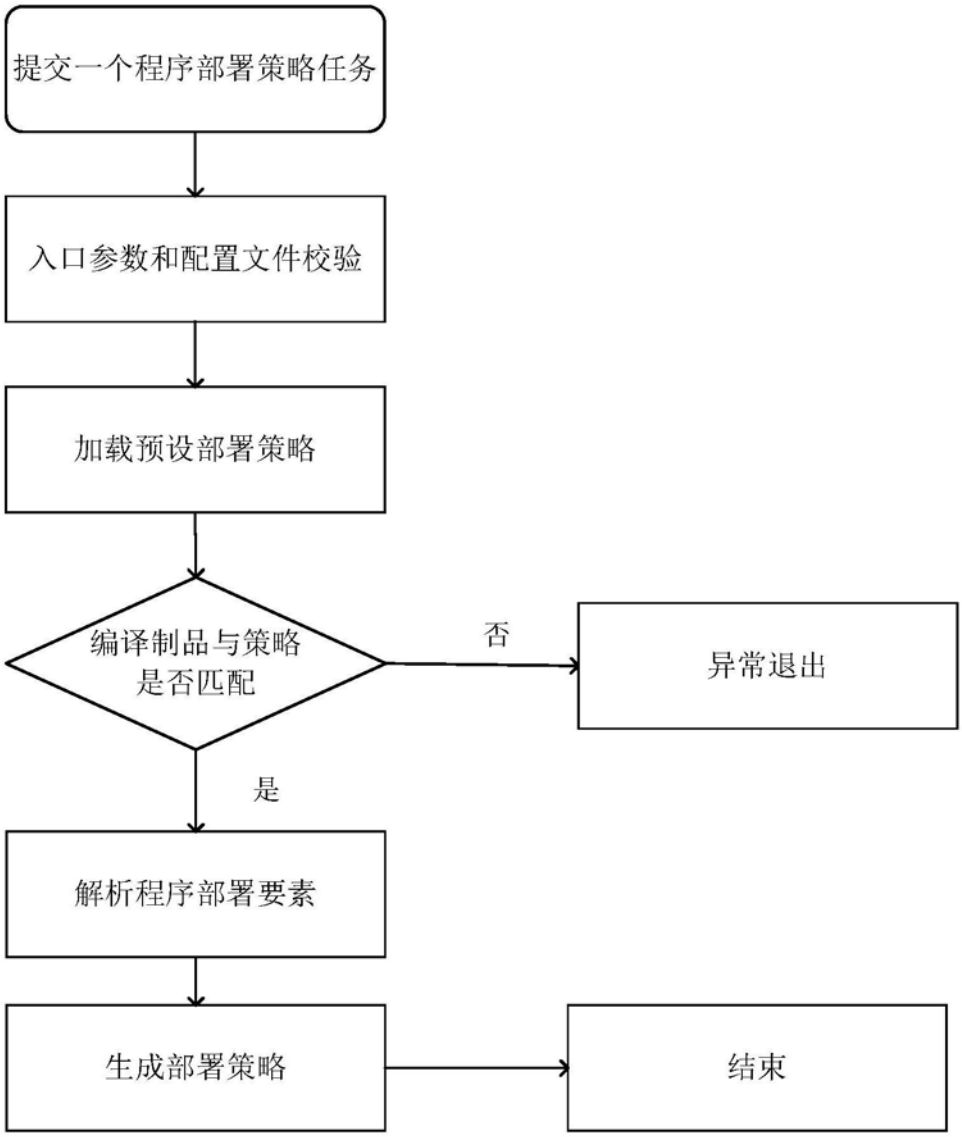

提交一个程序部署策略任务
入口参数和配置文件校验
加载预设部署策略
编译制品与策略是否匹配
否
异常退出
是
解析程序部署要素
生成部署策略
结束

图3

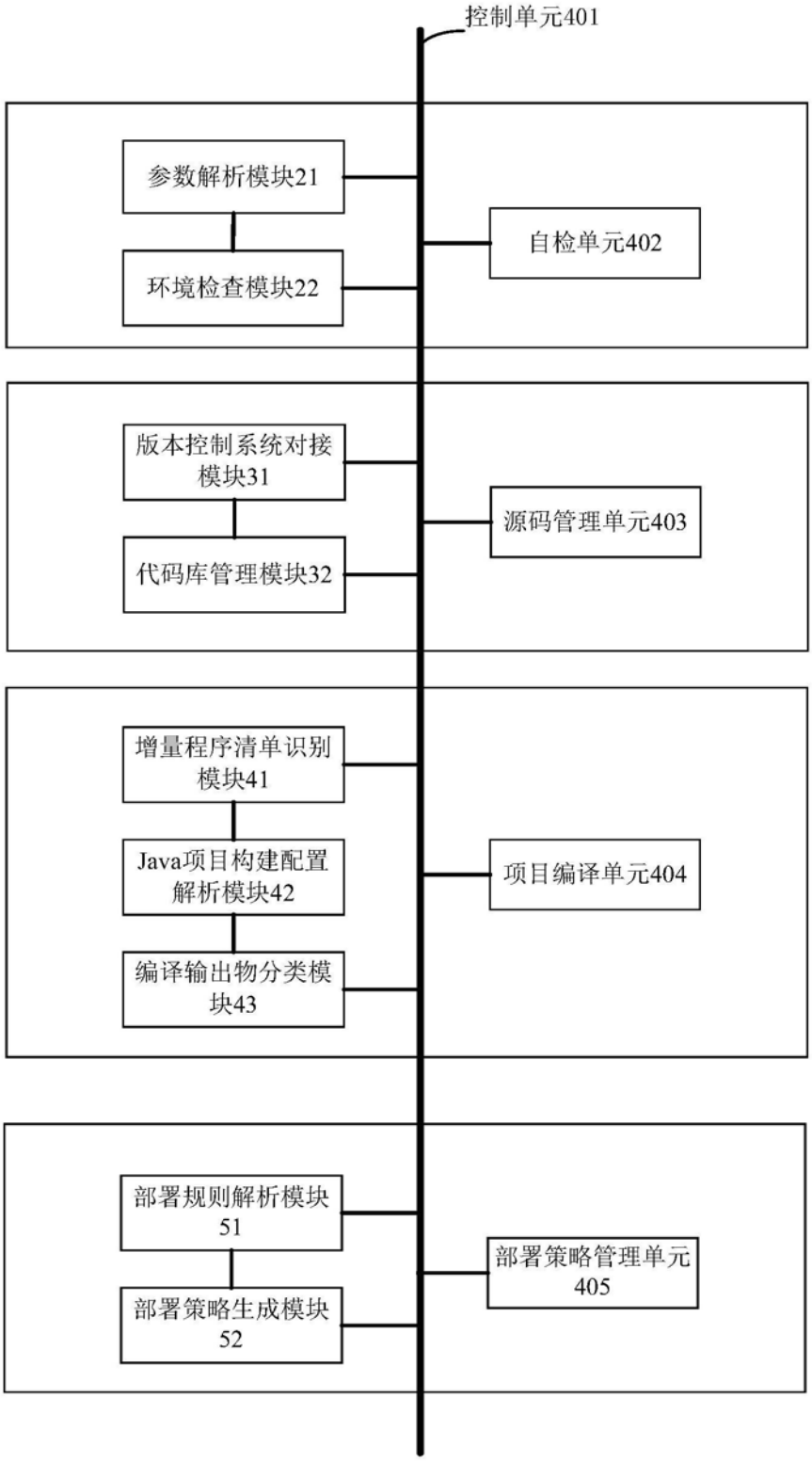

控制单元401
参数解析模块21
环境检查模块22
自检单元402
版本控制系统对接模块31
代码库管理模块32
源码管理单元403
增量程序清单识别模块41
Java项目构建配置解析模块42
编译输出物分类模块43
项目编译单元404
部署规则解析模块51
部署策略生成模块52
部署策略管理单元405

图4

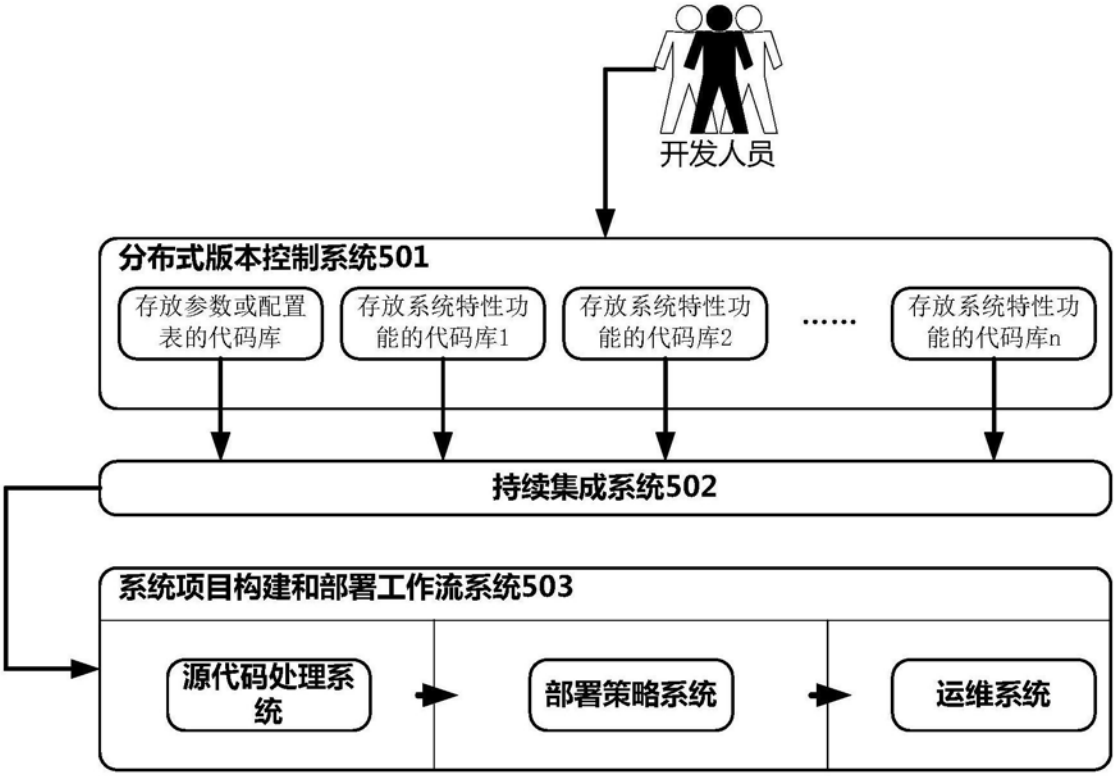

开发人员
分布式版本控制系统501
存放参数或配置表的代码库
存放系统特性功能的代码库1
存放系统特性功能的代码库2
.....
存放系统特性功能的代码库n
持续集成系统502
系统项目构建和部署工作流系统503
源代码处理系统
部署策略系统
运维系统

图5

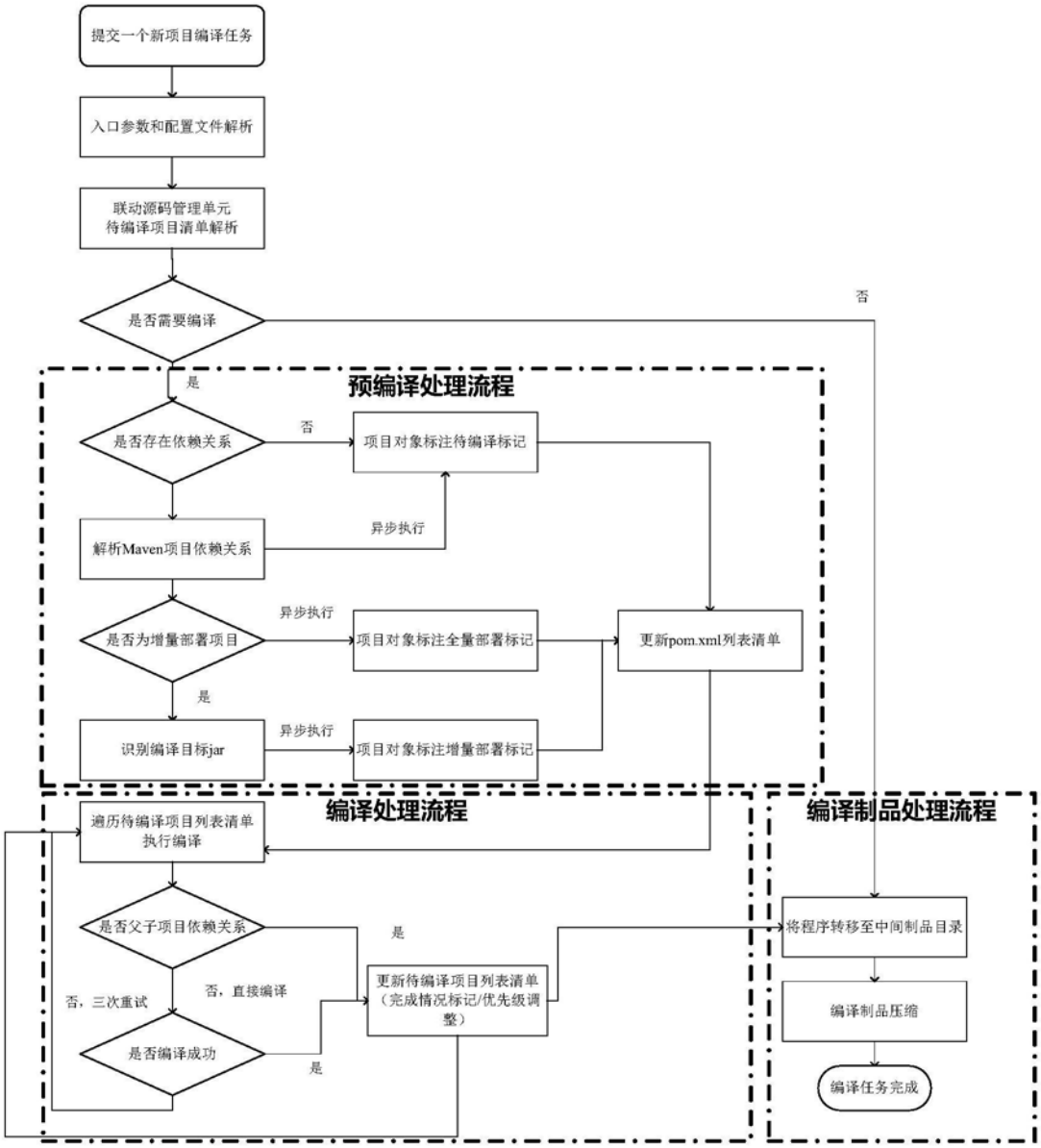

提交一个新项目编译任务
入口参数和配置文件解析
联动源码管理单元
待编译项目清单解析
是否需要编译
否
是
预编译处理流程
是否存在依赖关系
否
项目对象标注待编译标记
解析Maven项目依赖关系
异步执行
是否为增量部署项目
异步执行
项目对象标注全量部署标记
更新pom.xml列表清单
是
识别编译目标jar
异步执行
项目对象标注增量部署标记
编译处理流程
遍历待编译项目列表清单
执行编译
是否父子项目依赖关系
是
否，三次重试
否，直接编译
更新待编译项目列表清单
（完成情况标记/优先级调
整）
是否编译成功
是
编译制品处理流程
将程序转移至中间制品目录
编译制品压缩
编译任务完成

图6

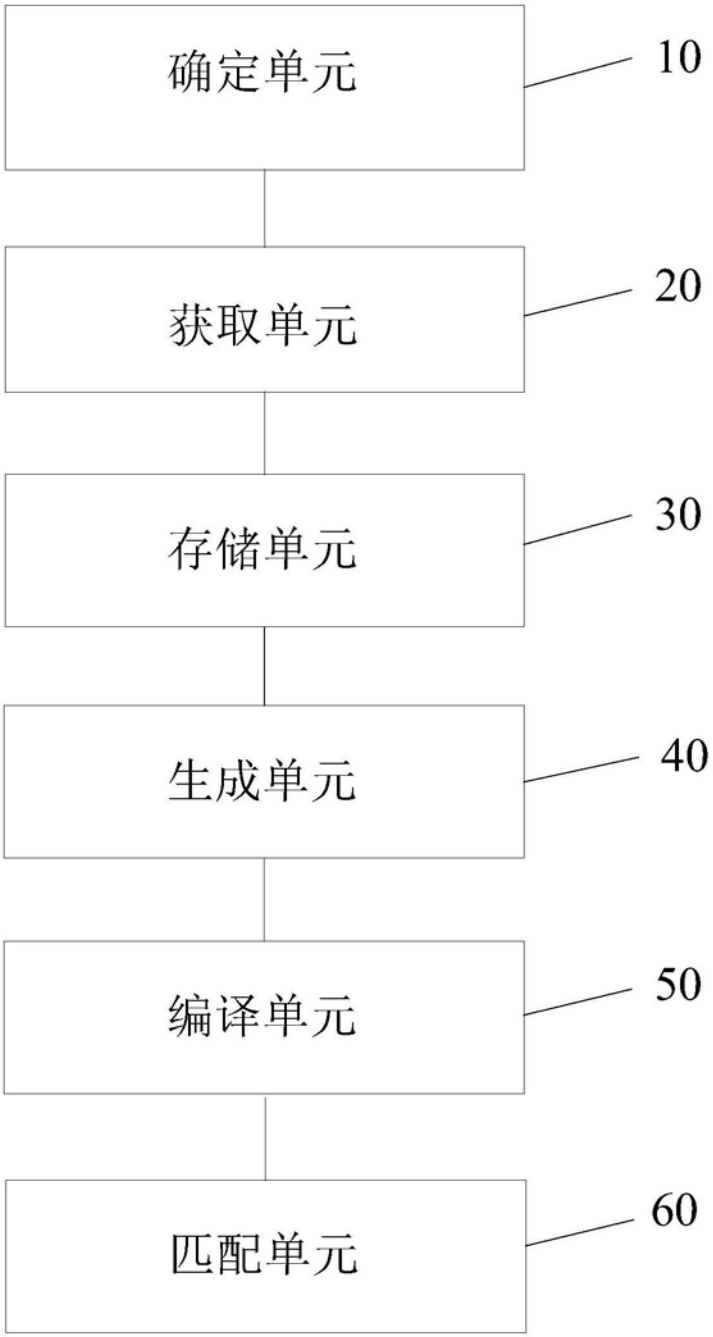


图7

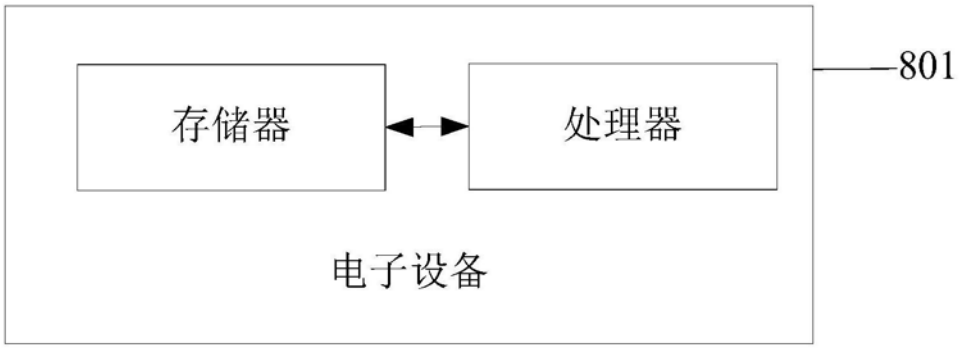


图8