



(12) 发明专利申请

(10) 申请公布号 CN 114020305 A

(43) 申请公布日 2022. 02. 08

(21) 申请号 202111063057.2

(22) 申请日 2021.09.10

(71) 申请人 天聚地合(苏州)数据股份有限公司  
地址 215128 江苏省苏州市苏州工业园区  
启月街288号紫金东方大厦307室

(72) 发明人 左磊 安蕴岳

(74) 专利代理机构 北京德崇智捷知识产权代理  
有限公司 11467

代理人 王欣

(51) Int. Cl.

G06F 8/654 (2018.01)

G06F 16/27 (2019.01)

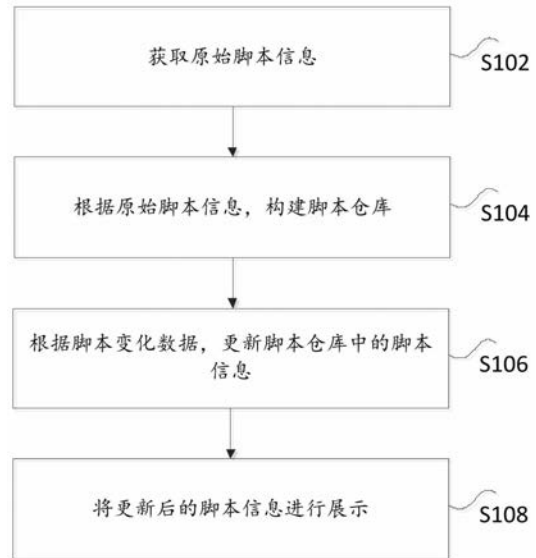
权利要求书2页 说明书8页 附图1页

(54) 发明名称

一种脚本数据配置方法及装置

(57) 摘要

本发明公开了一种脚本数据配置方法及装置。其中,该方法包括:获取原始脚本信息,其中,所述原始脚本信息包括:脚本标识、脚本数据,其中,所述脚本标识用于标识脚本的属性和变化信息;根据所述原始脚本信息,构建脚本仓库,其中,所述根据所述原始脚本信息,构建脚本仓库包括:获取所述脚本仓库空间信息;根据所述脚本仓库空间信息以及所述原始脚本信息,生成脚本存储位置;根据所述脚本存储位置构建所述脚本仓库;根据脚本变化数据,更新所述脚本仓库中的脚本信息;将更新后的所述脚本信息进行展示。本发明解决了现有技术中对于同一类型的项目,脚本操作基本相同,环境变量可能不同每一个项目定义一个部署脚本,需要修改时,根据项目情况修改,如果有大量项目需要修改,则必须逐个修改,即使改动的内容完全一致现有技术方案需要人力去维护,不能做到自动化处理的技术问题。



1. 一种脚本数据配置方法,其特征在于,包括:

获取原始脚本信息,其中,所述原始脚本信息包括:脚本标识、脚本数据,其中,所述脚本标识用于标识脚本的属性和变化信息;

根据所述原始脚本信息,构建脚本仓库,其中,所述根据所述原始脚本信息,构建脚本仓库包括:获取所述脚本仓库空间信息;根据所述脚本仓库空间信息以及所述原始脚本信息,生成脚本存储位置;根据所述脚本存储位置构建所述脚本仓库;

根据脚本变化数据,更新所述脚本仓库中的脚本信息;

将更新后的所述脚本信息进行展示。

2. 根据权利要求1所述的方法,其特征在于,在所述获取原始脚本信息之前,所述方法还包括:

获取项目激活状态,其中,所述项目激活状态包括:已激活,未激活。

3. 根据权利要求1所述的方法,其特征在于,在所述根据脚本变化数据,更新所述脚本仓库中的脚本信息之前,所述方法还包括:

获取项目的所述脚本变化数据。

4. 根据权利要求1所述的方法,其特征在于,所述根据脚本变化数据,更新所述脚本仓库中的脚本信息包括:

将所述脚本仓库中的所述原始脚本信息拆分为通用脚本信息和其他脚本信息;

根据所述脚本变化数据和同步程序,更新所述通用脚本信息。

5. 根据权利要求4所述的方法,其特征在于,所述将所述脚本仓库中的所述原始脚本信息拆分为通用脚本信息和其他脚本信息包括:

所述同步程序通过占位符来拆分所述原始脚本信息为所述通用脚本信息和所述其他脚本信息。

6. 一种脚本数据配置装置,其特征在于,包括:

获取模块,用于获取原始脚本信息,其中,所述原始脚本信息包括:脚本标识、脚本数据,其中,所述脚本标识用于标识脚本的属性和变化信息;

构建模块,用于根据所述原始脚本信息,构建脚本仓库,其中,所述根据所述原始脚本信息,构建脚本仓库包括:获取所述脚本仓库空间信息;根据所述脚本仓库空间信息以及所述原始脚本信息,生成脚本存储位置;根据所述脚本存储位置构建所述脚本仓库;

更新模块,用于根据脚本变化数据,更新所述脚本仓库中的脚本信息;

展示模块,用于将更新后的所述脚本信息进行展示。

7. 根据权利要求6所述的装置,其特征在于,所述装置还包括:

激活模块,用于获取项目激活状态,其中,所述项目激活状态包括:已激活,未激活。

8. 根据权利要求6所述的装置,其特征在于,所述装置还包括:

变化模块,用于获取项目的所述脚本变化数据。

9. 根据权利要求6所述的装置,其特征在于,所述更新模块包括:

拆分单元,用于将所述脚本仓库中的所述原始脚本信息拆分为通用脚本信息和其他脚本信息;

更新单元,用于根据所述脚本变化数据和同步程序,更新所述通用脚本信息。

10. 根据权利要求9所述的装置,其特征在于,所述拆分单元包括:

占位拆分子单元,所述同步程序通过占位符来拆分所述原始脚本信息为所述通用脚本信息和所述其他脚本信息。

11.一种非易失性存储介质,其特征在于,所述非易失性存储介质包括存储的程序,其中,所述程序运行时控制非易失性存储介质所在的设备执行权利要求1至5中任意一项所述的方法。

12.一种电子装置,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5中任意一项所述的方法。

## 一种脚本数据配置方法及装置

### 技术领域

[0001] 本发明涉及脚本数据处理领域,具体而言,涉及一种脚本数据配置方法及装置。

### 背景技术

[0002] 随着智能化科技的不断发展,人们的生活、工作、学习之中越来越多地用到了智能化设备,使用智能化科技手段,提高了人们的生活的质量,增加了人们学习和工作的效率。

[0003] 目前,在项目的持续集成过程中,会为每一个项目定义一个部署脚本,每个项目的类型不同,脚本操作也不同。但对于同一类型的项目,脚本操作基本相同,环境变量可能不同。每一个项目定义一个部署脚本,需要修改时,根据项目情况修改,如果有大量项目需要修改,则必须逐个修改,即使改动的内容完全一致。现有技术方案需要人力去维护,不能做到自动化。

[0004] 针对上述的问题,目前尚未提出有效的解决方案。

### 发明内容

[0005] 本发明实施例提供了一种脚本数据配置方法及装置,以至少解决现有技术中对于同一类型的项目,脚本操作基本相同,环境变量可能不同每一个项目定义一个部署脚本,需要修改时,根据项目情况修改,如果有大量项目需要修改,则必须逐个修改,即使改动的内容完全一致现有技术方案需要人力去维护,不能做到自动化处理的技术问题。

[0006] 根据本发明实施例的一个方面,提供了一种脚本数据配置方法,包括:获取原始脚本信息,其中,所述原始脚本信息包括:脚本标识、脚本数据,其中,所述脚本标识用于标识脚本的属性和变化信息;根据所述原始脚本信息,构建脚本仓库,其中,所述根据所述原始脚本信息,构建脚本仓库包括:获取所述脚本仓库空间信息;根据所述脚本仓库空间信息以及所述原始脚本信息,生成脚本存储位置;根据所述脚本存储位置构建所述脚本仓库;根据脚本变化数据,更新所述脚本仓库中的脚本信息;将更新后的所述脚本信息进行展示。

[0007] 可选的,在所述获取原始脚本信息之前,所述方法还包括:获取项目激活状态,其中,所述项目激活状态包括:已激活,未激活。

[0008] 可选的,在所述根据脚本变化数据,更新所述脚本仓库中的脚本信息之前,所述方法还包括:获取项目的所述脚本变化数据。

[0009] 可选的,所述根据脚本变化数据,更新所述脚本仓库中的脚本信息包括:将所述脚本仓库中的所述原始脚本信息拆分为通用脚本信息和其他脚本信息;根据所述脚本变化数据和同步程序,更新所述通用脚本信息。

[0010] 可选的,所述将所述脚本仓库中的所述原始脚本信息拆分为通用脚本信息和其他脚本信息包括:所述同步程序通过占位符来拆分所述原始脚本信息为所述通用脚本信息和所述其他脚本信息。

[0011] 根据本发明实施例的另一方面,还提供了一种脚本数据配置装置,包括:获取模块,用于获取原始脚本信息;构建模块,用于根据所述原始脚本信息,构建脚本仓库;更新模

块,用于根据脚本变化数据,更新所述脚本仓库中的脚本信息;展示模块,用于将更新后的所述脚本信息进行展示。

[0012] 可选的,所述装置还包括:激活模块,用于获取项目激活状态,其中,所述项目激活状态包括:已激活,未激活。

[0013] 可选的,所述装置还包括:变化模块,用于获取项目的所述脚本变化数据。

[0014] 可选的,所述更新模块包括:拆分单元,用于将所述脚本仓库中的所述原始脚本信息拆分为通用脚本信息和其他脚本信息;更新单元,用于根据所述脚本变化数据和同步程序,更新所述通用脚本信息。

[0015] 可选的,所述拆分单元包括:占位拆分子单元,所述同步程序通过占位符来拆分所述原始脚本信息为所述通用脚本信息和所述其他脚本信息。

[0016] 根据本发明实施例的另一方面,还提供了一种非易失性存储介质,所述非易失性存储介质包括存储的程序,其中,所述程序运行时控制非易失性存储介质所在的设备执行一种脚本数据配置方法。

[0017] 根据本发明实施例的另一方面,还提供了一种电子装置,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一种脚本数据配置方法。

[0018] 在本发明实施例中,采用获取原始脚本信息;根据所述原始脚本信息,构建脚本仓库;根据脚本变化数据,更新所述脚本仓库中的脚本信息;将更新后的所述脚本信息进行展示的方式,解决了现有技术中对于同一类型的项目,脚本操作基本相同,环境变量可能不同每一个项目定义一个部署脚本,需要修改时,根据项目情况修改,如果有大量项目需要修改,则必须逐个修改,即使改动的内容完全一致现有技术需要人力去维护,不能做到自动化处理的技术问题。

## 附图说明

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

图1是根据本发明实施例的一种脚本数据配置方法的流程图;

图2是根据本发明实施例的一种脚本数据配置装置的结构框图。

## 具体实施方式

[0020] 为了使本技术领域的人员更好地理解本发明方案,下面将结合本发明实施例中的附图,对本发明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发明一部分的实施例,而不是全部的实施例。基于本发明中的实施例,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,都应当属于本发明保护的范围。

[0021] 需要说明的是,本发明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,以便这里描述的本发明的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆

盖不排除的包含,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。

[0022] 根据本发明实施例,提供了一种脚本数据配置方法的方法实施例,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。

[0023] 实施例一

图1是根据本发明实施例的一种脚本数据配置方法的流程图,如图1所示,该方法包括如下步骤:

步骤S102,获取原始脚本信息。

[0024] 具体的,本发明实施例为了实现将脚本信息进行更新部署,并将更新结果同步到脚本仓库的所有脚本中,以解决现有技术中对于同一类型的项目,脚本操作基本相同,环境变量可能不同每一个项目定义一个部署脚本,需要修改时,根据项目情况修改,如果有大量项目需要修改,则必须逐个修改,即使改动的内容完全一致现有技术方案需要人力去维护,不能做到自动化处理的技术问题,首先需要获取原始的部署脚本信息,其中,原始的部署脚本信息是通过相应的项目情况来进行采集的,脚本信息中包含了项目涉及到的脚本参数等数据,然后根据通过用户传输或采集器采集到的原始脚本信息构建后续的脚本信息集合体,从而为脚本的更新处理提供前提条件。

[0025] 可选的,在所述获取原始脚本信息之前,所述方法还包括:获取项目激活状态,其中,所述项目激活状态包括:已激活,未激活。

[0026] 具体的,为了增加本发明实施例的执行效率,节省不必要的计算资源,在所述获取原始脚本信息之前,所述方法还包括:获取项目激活状态,其中,所述项目激活状态包括:已激活,未激活。其中,项目激活状态的获取可以通过状态采集器对项目是否已经开始进行而采集,如果项目处于激活状态,那么才可以对后续原始脚本信息和更新信息进行采集和获取,以便节省计算资源,增加计算的效率,从而达到优化整个项目更新的技术目的。

[0027] 步骤S104,根据所述原始脚本信息,构建脚本仓库。

[0028] 具体的,在本发明实施例获取到了原始脚本信息之后,需要根据获取到的原始脚本信息来构建脚本仓库,其中,脚本仓库项目存储了不同类型项目的部署脚本,由专人负责维护这些部署脚本。若脚本发生变化,自动同步到各个项目。脚本仓库是若干脚本信息汇总后的数据库,存储着所有的脚本信息,便于后续进行脚本信息的更新和同步,以及调用。例如,该项目包含的关键文件:deploy\_laravel.yml;deploy\_docker.yml;deploy\_go.yml;deploy\_python.yml;deploy\_update.sh;其中:deploy\_laravel.yml 用于部署Laravel项目;deploy\_go.yml 用于部署Go项目;deploy\_docker.yml 用于部署Docker项目;deploy\_python.yml 用于部署Python项目;deploy\_update.sh 定义了同步以上文件的脚本,在本发明实施例中称为“同步程序”。

[0029] 步骤S106,根据脚本变化数据,更新所述脚本仓库中的脚本信息。

[0030] 具体的,当需要对上述实施例中生成的脚本仓库中的脚本信息进行信息更新的时候,即当发生了脚本变化数据之后,需要将脚本变化的数据通过同步执行程序同步到相应

的脚本中,并进行同步,以便后续对任何脚本实施项目调取的时候,都可以直接获取更新后的脚本。

[0031] 例如,项目类型有多种,如:PHP项目、前端项目、APP项目、小程序项目等。每个项目都有其部署脚本,部署脚本定义了一套部署的流程,不同类型的项目流程不完全相同。这些部署脚本并不是一成不变的,根据实际需要,可以随时变更,但必须由管理员负责变更。有一个部署脚本仓库,存储了不同类型的项目的部署脚本,文件结构大致如下:PHP项目的部署脚本1.0、前端项目的部署脚本1.1、小程序项目的部署脚本1.2。通过以下案例来说明:管理员确定了部署脚本,将对应的部署脚本存储在脚本仓库中,当前仓库中的文件列表如下:PHP项目部署脚本1.0、APP项目部署脚本1.1、小程序项目部署脚本1.5。初始化阶段,同步程序将各部署脚本同步到各项目,如:将《PHP项目部署脚本1.0》同步到PHP项目中、将《APP项目部署脚本1.1》同步到APP项目中、将《小程序项目部署脚本1.5》同步到小程序项目中。

[0032] 另外,当管理员决定改动部署脚本时,如:PHP项目部署脚本由1.0升级到1.1(增加了清理缓存流程)、APP项目部署脚本由1.1升级到1.2(增加了重启流程)。同步程序会执行以下操作:将《PHP项目部署脚本1.1》同步到PHP项目中、将《APP项目部署脚本1.2》同步到APP项目中。部署脚本并不是一成不变的,当其变化时,会自动同步变动的内容到相应的项目中。对于本发明实施例的同步程序,可以通过以下方式来运行:获取脚本仓库里的脚本列表、针对每个脚本,都进行接下来的操作,以PHP项目部署脚本为例、调用代码仓库提供的API服务,找出所有的PHP项目、下载上一步筛选的PHP项目、将PHP项目脚本数据配置到下载的每个PHP项目中、提交每个PHP项目的变动到代码仓库。

[0033] 可选的,在所述根据脚本变化数据,更新所述脚本仓库中的脚本信息之前,所述方法还包括:获取项目的所述脚本变化数据。可选的,所述根据脚本变化数据,更新所述脚本仓库中的脚本信息包括:将所述脚本仓库中的所述原始脚本信息拆分为通用脚本信息和其他脚本信息;根据所述脚本变化数据和同步程序,更新所述通用脚本信息。可选的,所述将所述脚本仓库中的所述原始脚本信息拆分为通用脚本信息和其他脚本信息包括:所述同步程序通过占位符来拆分所述原始脚本信息为所述通用脚本信息和所述其他脚本信息。具体的,在本发明实施例中,可以为脚本仓库项目定义一个Webhook,当脚本仓库项目内的文件发生变化时,执行同步程序。执行日志举例:开始执行同步程序;

```
-----  
    当前项目: www.abc.com;  
    项目类型:laravel;  
    比较脚本仓库中的deploy_laravel.yml文件和www.abc.com的目录中的deploy_  
laravel.yml文件;  
    工作流程文件无变化,无需继续处理。
```

```
[0034] -----  
    当前项目:xyz.abc.com;  
    项目类型:docker;  
    比较脚本仓库中的deploy_docker.yml文件和xyz.abc.com的目录中的deploy_  
docker.yml文件;  
    工作流程文件有变化;      提交变化到xyz.abc.com的代码仓库。
```



激活,未激活。其中,项目激活状态的获取可以通过状态采集器对项目是否已经开始进行而采集,如果项目处于激活状态,那么才可以对后续的原始脚本信息和更新信息进行采集和获取,以便节省计算资源,增加计算的效率,从而达到优化整个项目更新的技术目的。

[0044] 构建模块22,用于根据所述原始脚本信息,构建脚本仓库。

[0045] 具体的,在本发明实施例获取到了原始脚本信息之后,需要根据获取到的原始脚本信息来构建脚本仓库,其中,脚本仓库项目存储了不同类型项目的部署脚本,由专人负责维护这些部署脚本。若脚本发生变化,自动同步到各个项目。脚本仓库是若干脚本信息汇总后的数据库,存储着所有的脚本信息,便于后续进行脚本信息的更新和同步,以及调用。例如,该项目包含的关键文件:deploy\_laravel.yml;deploy\_docker.yml;deploy\_go.yml;deploy\_python.yml;deploy\_update.sh;其中:deploy\_laravel.yml 用于部署Laravel项目;deploy\_go.yml 用于部署Go项目;deploy\_docker.yml 用于部署Docker项目;deploy\_python.yml 用于部署Python项目;deploy\_update.sh 定义了同步以上文件的脚本,在本发明实施例中称为“同步程序”。

[0046] 更新模块24,用于根据脚本变化数据,更新所述脚本仓库中的脚本信息。

[0047] 具体的,当需要对上述实施例中生成的脚本仓库中的脚本信息进行信息更新的时候,即当发生了脚本变化数据之后,需要将脚本变化的数据通过同步执行程序同步到相应的脚本中,并进行同步,以便后续对任何脚本实施项目调取的时候,都可以直接获取更新后的脚本。

[0048] 例如,项目类型有多种,如:PHP项目、前端项目、APP项目、小程序项目等。每个项目都有其部署脚本,部署脚本定义了一套部署的流程,不同类型的项目流程不完全相同。这些部署脚本并不是一成不变的,根据实际需要,可以随时变更,但必须由管理员负责变更。有一个部署脚本仓库,存储了不同类型的项目的部署脚本,文件结构大致如下:PHP项目的部署脚本1.0、前端项目的部署脚本1.1、小程序项目的部署脚本1.2。通过以下案例来说明:管理员确定了部署脚本,将对应的部署脚本存储在脚本仓库中,当前仓库中的文件列表如下:PHP项目部署脚本1.0、APP项目部署脚本1.1、小程序项目部署脚本1.5。初始化阶段,同步程序将各部署脚本同步到各项目,如:将《PHP项目部署脚本1.0》同步到PHP项目中、将《APP项目部署脚本1.1》同步到APP项目中、将《小程序项目部署脚本1.5》同步到小程序项目中。

[0049] 另外,当管理员决定改动部署脚本时,如:PHP项目部署脚本由1.0升级到1.1(增加了清理缓存流程)、APP项目部署脚本由1.1升级到1.2(增加了重启流程)。同步程序会执行以下操作:将《PHP项目部署脚本1.1》同步到PHP项目中、将《APP项目部署脚本1.2》同步到APP项目中。部署脚本并不是一成不变的,当其变化时,会自动同步变动的内容到相应的项目中。对于本发明实施例的同步程序,可以通过以下方式来运行:获取脚本仓库里的脚本列表、针对每个脚本,都进行接下来的操作,以PHP项目部署脚本为例、调用代码仓库提供的API服务,找出所有的PHP项目、下载上一步筛选的PHP项目、将PHP项目脚本数据配置到下载的每个PHP项目中、提交每个PHP项目的变动到代码仓库。

[0050] 可选的,所述装置还包括:变化模块,用于获取项目的所述脚本变化数据。

[0051] 可选的,所述更新模块包括:拆分单元,用于将所述脚本仓库中的所述原始脚本信息拆分为通用脚本信息和其他脚本信息;更新单元,用于根据所述脚本变化数据和同步程序,更新所述通用脚本信息。



[0060] 具体的,上述方法包括:获取原始脚本信息;根据所述原始脚本信息,构建脚本仓库;根据脚本变化数据,更新所述脚本仓库中的脚本信息;将更新后的所述脚本信息进行展示。

[0061] 根据本发明实施例的另一方面,还提供了一种电子装置,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一种脚本数据配置方法。

[0062] 具体的,上述方法包括:获取原始脚本信息;根据所述原始脚本信息,构建脚本仓库;根据脚本变化数据,更新所述脚本仓库中的脚本信息;将更新后的所述脚本信息进行展示。

[0063] 通过上述实施例,解决了现有技术中对于同一类型的项目,脚本操作基本相同,环境变量可能不同每一个项目定义一个部署脚本,需要修改时,根据项目情况修改,如果有大量项目需要修改,则必须逐个修改,即使改动的内容完全一致现有技术需要人力去维护,不能做到自动化处理的技术问题。

[0064] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0065] 在本发明的上述实施例中,对各个实施例的描述都各有侧重,某个实施例中沒有详述的部分,可以参见其他实施例的相关描述。

[0066] 在本申请所提供的几个实施例中,应该理解到,所揭露的技术内容,可通过其它的方式实现。其中,以上所描述的装置实施例仅仅是示意性的,例如所述单元的划分,可以为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,单元或模块的间接耦合或通信连接,可以是电性或其它的形式。

[0067] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0068] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0069] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时,可以存储在一个计算机可读存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可为个人计算机、服务器或者网络设备)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0070] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

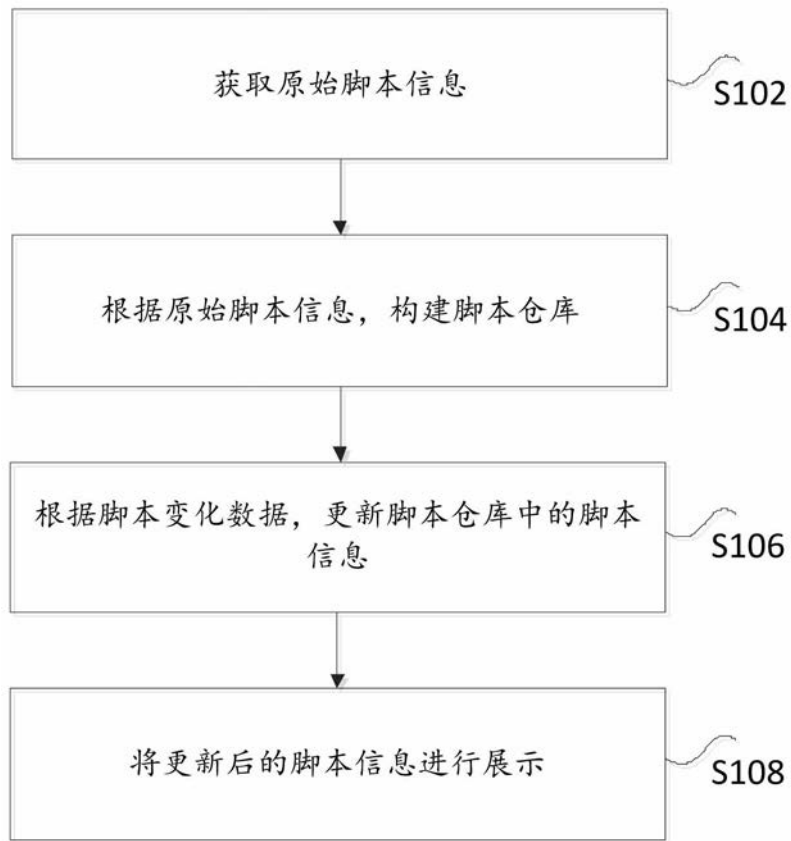


图1



图2