



(12)发明专利申请

(10)申请公布号 CN 108874409 A

(43)申请公布日 2018. 11. 23

(21)申请号 201710321989.X

(22)申请日 2017.05.09

(71)申请人 腾讯科技(深圳)有限公司

地址 518000 广东省深圳市南山区高新区
科技中一路腾讯大厦35层

(72)发明人 武建芳

(74)专利代理机构 深圳市深佳知识产权代理事
务所(普通合伙) 44285

代理人 王仲凯

(51)Int. Cl.

G06F 8/65(2018.01)

G06F 8/71(2018.01)

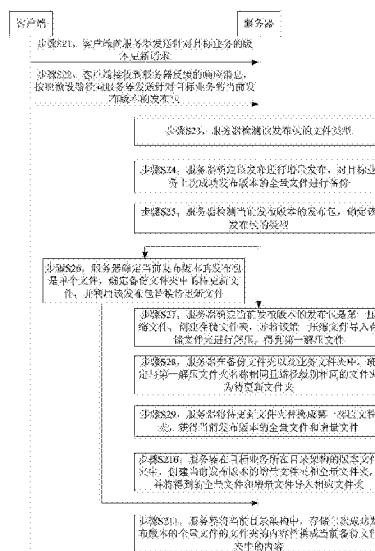
权利要求书3页 说明书14页 附图7页

(54)发明名称

一种信息更新方法、装置、服务器及系统

(57)摘要

本申请提供了一种信息更新方法、装置、服务器及系统,服务器通过解析客户端上传的针对目标业务的当前发布版本的发布包的文件类型,确定对该发布包进行增量发布时,可以将该目标业务的当前目录结构反馈至客户端输出,使开发人员具体确定该发布包目标上传路径后,由服务器按照该目标上传路径将发布包导入当前目录结构的对应文件夹,从而利用导入的发布包更新该目标业务上次成功发布版本的全量文件,得到当前发布版本的全量文件。可见,开发人员能够将自己负责的目标业务的功能部分的发布包直接上传并发布,无需与该目标业务的其他开发人员进行配合,非常方便,且减少了上传数据量,降低了网络耗时,大大提高了信息更新效率。



1. 一种信息更新方法,其特征在于,所述方法包括:
 - 接收客户端上传的针对目标业务的当前发布版本的发布包;
 - 检测所述发布包的文件类型,确定对所述发布包进行增量发布;
 - 将所述目标业务的当前目录结构反馈至所述客户端输出,以使所述客户端基于所述当前目录结构确定所述发布包的目标上传路径;
 - 按照所述目标上传路径,将所述发布包导入所述当前目录结构的对应文件夹;
 - 利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,得到当前发布版本的全量文件。
2. 根据权利要求1所述的方法,其特征在于,在所述确定对所述发布包进行增量发布之后,所述方法还包括:
 - 对所述目标业务上次成功发布版本的全量文件进行备份;
 - 所述利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,包括:
 - 利用导入的所述的发布包,更新得到的备份文件夹,所述备份文件夹包含在所述目标业务所在的目录架构中;
 - 利用更新后的备份文件夹内容,替换所述目录架构中用于存储上次成功发布版本的全量文件的文件夹内容。
3. 根据权利要求1所述的方法,其特征在于,确定所述发布包是单个文件,所述按照所述目标上传路径,将所述发布包导入所述当前目录结构的对应文件夹,包括:
 - 按照所述目标上传路径,将所述单个文件导入所述当前目录结构的对应文件夹;
 - 所述利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,包括:
 - 利用导入的所述单个文件,替换所述目标业务上次成功发布版本的全量文件中的对应文件。
4. 根据权利要求1所述的方法,其特征在于,确定所述发布包是第一压缩文件,所述方法还包括:
 - 在所述目标业务所在的目录架构中创建存储文件夹;
 - 所述利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,包括:
 - 将导入的第一压缩文件发送至所述存储文件夹进行解压,得到相应的第一解压文件夹;
 - 查询所述目录架构下与所述第一解压文件夹名称相同,且路径级别相同的文件夹,并将查询到的文件夹确定为待更新文件夹;
 - 将所述待更新文件夹替换成所述第一解压文件夹,获得当前发布版本的全量文件。
5. 根据权利要求1所述的方法,其特征在于,在将所述发布包导入所述当前目录结构的对应文件夹时,所述方法还包括:
 - 生成所述目标业务的增量文件;
 - 在所述目标业务所在目录架构的版本文件夹中,创建当前发布版本对应全量文件夹和增量文件夹;
 - 将得到的所述当前发布版本的全量文件导入所述全量文件夹,并将得到的所述增量文件导入所述增量文件夹。
6. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

当所述发布包是第二压缩文件,确定对所述发布包进行全量发布;

对所述第二压缩文件进行解压;

利用解压得到的文件或文件夹,生成所述目标业务的目录结构,以及用于存储当前发布版本发布包的文件夹的目录结构;

将解压得到的文件或文件夹导入生成的所述文件夹的目录结构中的相应文件夹,获得当前发布版本的全量文件;

在所述目标业务所在目录架构的版本文件夹中,创建当前发布版本对应全量文件夹,并将获得的全量文件导入所述全量文件夹。

7. 根据权利要求2所述的方法,其特征在于,所述对所述目标业务上次成功发布版本的全量文件进行备份,包括:

将所述目标业务所在目录架构中,用于临时存储当前发布版本的全量文件的文件夹确定为备份文件夹;

删除所述备份文件夹中的内容,并将用于存储上次成功发布版本的全量文件的文件夹内容拷贝至所述备份文件夹。

8. 一种信息更新装置,其特征在于,所述装置包括:

数据传输模块,用于接收客户端上传的针对目标业务的当前发布版本的发布包;

第一数据处理模块,用于解析所述发布包的文件类型,确定对所述发布包进行增量发布,触发所述数据传输模块将所述目标业务的当前目录结构反馈至所述客户端输出,以使所述客户端基于所述当前目录结构确定所述发布包的目标上传路径;

第一数据导入模块,用于按照所述目标上传路径,将所述发布包导入所述当前目录结构的对应文件夹;

文件更新模块,用于利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,得到当前发布版本的全量文件。

9. 根据权利要求8所述的装置,其特征在于,所述装置还包括:

文件备份模块,用于对所述目标业务上次成功发布版本的全量文件进行备份;

所述文件更新模块具体用于利用导入的所述的发布包,更新得到的备份文件夹,所述备份文件夹包含在所述目标业务所在的目录架构中,并利用更新后的备份文件夹内容,替换所述目录架构中用于存储上次成功发布版本的全量文件的文件夹内容。

10. 根据权利要求8所述的装置,其特征在于,当所述数据处理模块确定所述发布包是第一压缩文件,所述装置还包括:

第一创建模块,用于在所述目标业务所在的目录架构中创建存储文件夹;

所述文件更新模块具体用于将导入的第一压缩文件发送至所述存储文件夹进行解压,得到相应的第一解压文件夹,查询所述目录架构下与所述第一解压文件夹名称相同,且路径级别相同的文件夹,并将查询到的文件夹确定为待更新文件夹,将所述待更新文件夹替换成所述第一解压文件夹,获得当前发布版本的全量文件。

11. 根据权利要求8所述的装置,其特征在于,所述装置还包括:

第一生成模块,用于在将所述发布包导入所述当前目录结构的对应文件夹时,生成所述目标业务的增量文件;

第二创建模块,用于在所述目标业务所在目录架构的版本文件夹中,创建当前发布版

本对应全量文件夹和增量文件夹；

第二数据导入模块，用于将得到的所述当前发布版本的全量文件导入所述全量文件夹，并将得到的所述增量文件导入所述增量文件夹。

12. 根据权利要求8所述的装置，其特征在于，所述装置还包括：

第二数据处理模块，用于当所述发布包是第二压缩文件，确定对所述发布包进行全量发布；

解压模块，用于对所述第二压缩文件进行解压；

第二生成模块，用于利用解压得到的文件或文件夹，生成所述目标业务的目录结构，以及用于存储当前发布版本发布包的文件夹的目录结构；

第三数据导入模块，用于将解压得到的文件或文件夹导入生成的所述文件夹的目录结构中的相应文件夹，获得当前发布版本的全量文件；

第三创建模块，用于在所述目标业务所在目录架构的版本文件夹中，创建当前发布版本对应全量文件夹，并将获得的全量文件导入所述全量文件夹。

13. 一种服务器，其特征在于，所述服务器包括：

通信接口，用于接收客户端上传的针对目标业务的当前发布版本的发布包；

存储器，用于保存多个指令；

处理器，用于加载并执行所述多个指令，包括：

检测所述发布包的文件类型，确定对所述发布包进行增量发布；

将所述目标业务的当前目录结构反馈至所述客户端输出，以使所述客户端基于所述当前目录结构确定所述发布包的目标上传路径；

按照所述目标上传路径，将所述发布包导入所述当前目录结构的对应文件夹；

利用导入的所述发布包，更新所述目标业务上次成功发布版本的全量文件，得到当前发布版本的全量文件。

14. 一种信息更新系统，其特征在于，所述系统包括：服务器以及至少一个客户端，其中：

所述客户端，用于按照预设路径，向所述服务器上传针对目标业务的当前发布版本的发布包；输出所述服务器反馈的所述目标业务的当前目录结构，并基于所述当前目录结构，确定所述发布包的目标上传路径；

所述服务器，用于解析所述发布包的文件类型，确定对所述发布包进行增量发布，并将所述目标业务的当前目录结构反馈至所述客户端，按照所述客户端发送的所述目标上传路径，将所述发布包导入所述当前目录结构的对应文件夹，并利用导入的所述发布包，更新所述目标业务上次成功发布版本的全量文件，得到当前发布版本的全量文件。

一种信息更新方法、装置、服务器及系统

技术领域

[0001] 本申请主要涉及互联网技术领域,更具体地说是涉及一种信息更新方法、装置、服务器及系统。

背景技术

[0002] 如今,随着计算机网络技术的快速发展,市面上出现了各种各样的应用软件,如各种办公软件、音视频软件、浏览器、购物软件、导航软件等等,给人们的工作和生活带来了极大便利。在实际应用中,应用软件通常会因为功能的改进和漏洞的修补,需要不断进行升级,这就需要软件发布者发布新版本。

[0003] 对此,现有技术对原版本软件进行升级时,通常是将新版本的软件发布包重新发布一次,可见,这种方式由于每次都需要生成新版本完整的发布包,然后上传到发布机器,再由发布机器发布至线上机器,非常耗时。并且,由于应用软件的开发生态通常是分为多个模块,每一个模块的开发和维护经常会分配给多个技术人员负责,在发布新版本时,需要将自己负责的修改部分与其他负责人的部分进行合并,才能得到新版本的发布包,完成新版本的发布,过程比较繁琐。

[0004] 综上,如何提高新版本发布的便携性,降低网络耗时成为技术人员重要研究方向。

发明内容

[0005] 有鉴于此,本申请提供了一种信息更新方法、装置、服务器及系统,不需要与目标业务的其他开发人员配合,就能够直接完成自己负责的目标业务功能部分的发布包的发布,非常方便,且减少了上传数据量,降低了网络耗时,大大提高了信息更新效率。

[0006] 为了实现上述目的,本申请提供了以下技术方案:

[0007] 本申请实施例还提供一种文件更新方法,接收客户端上传的针对目标业务的当前发布版本的发布包;

[0008] 检测所述发布包的文件类型,确定对所述发布包进行增量发布;

[0009] 将所述目标业务的当前目录结构反馈至所述客户端输出,以使所述客户端基于所述当前目录结构确定所述发布包的目标上传路径;

[0010] 按照所述目标上传路径,将所述发布包导入所述当前目录结构的对应文件夹;

[0011] 利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,得到当前发布版本的全量文件。

[0012] 本申请实施例还提供了一种装置,所述装置包括:

[0013] 数据传输模块,用于接收客户端上传的针对目标业务的当前发布版本的发布包;

[0014] 第一数据处理模块,用于解析所述发布包的文件类型,确定对所述发布包进行增量发布,触发所述数据传输模块将所述目标业务的当前目录结构反馈至所述客户端输出,以使所述客户端基于所述当前目录结构确定所述发布包的目标上传路径;

[0015] 第一数据导入模块,用于按照所述目标上传路径,将所述发布包导入所述当前目

录结构的对应文件夹；

[0016] 文件更新模块,用于利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,得到当前发布版本的全量文件。

[0017] 本申请实施例还提供一种服务器,所述服务器包括:

[0018] 通信接口,用于接收客户端上传的针对目标业务的当前发布版本的发布包;

[0019] 存储器,用于保存多个指令;

[0020] 处理器,用于加载并执行所述多个指令,包括:

[0021] 检测所述发布包的文件类型,确定对所述发布包进行增量发布;

[0022] 将所述目标业务的当前目录结构反馈至所述客户端输出,以使所述客户端基于所述当前目录结构确定所述发布包的目标上传路径;

[0023] 按照所述目标上传路径,将所述发布包导入所述当前目录结构的对应文件夹;

[0024] 利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,得到当前发布版本的全量文件。

[0025] 本申请实施例还提供一种信息更新系统,所述系统包括:服务器以及至少一个客户端,其中:

[0026] 所述客户端,用于按照预设路径,向所述服务器上传针对目标业务的当前发布版本的发布包;输出所述服务器反馈的所述目标业务的当前目录结构,并基于所述当前目录结构,确定所述发布包的目标上传路径;

[0027] 所述服务器,用于解析所述发布包的文件类型,确定对所述发布包进行增量发布,并将所述目标业务的当前目录结构反馈至所述客户端,按照所述客户端发送的所述目标上传路径,将所述发布包导入所述当前目录结构的对应文件夹,并利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,得到当前发布版本的全量文件。

[0028] 基于上述技术方案,本申请实施例提供了一种信息更新方法、装置、服务器和系统,通过客户端将针对目标业务的当前发布版本的发布包上传到服务器,服务器通过解析发布包的文件类型,确定出对该发布包进行增量发布时,可以将该目标业务的当前目录结构反馈至客户端输出,以便开发人员根据需要灵活且自助确定该发布包的目标上传路径,,从而使服务器按照该目标上传路径直接将发布包导入该当前目录结构的对应文件夹,利用导入的发布包更新该目标业务上次成功发布版本的全量文件,得到当前发布版本的全量文件。由此可见,本申请中,开发人员能够根据自己负责的目标业务的功能部分,灵活选择该发布包具体的上传路径,无需与该目标业务的其他开发人员开发的发布包进行配合,即可实现该发布包的增量发布,得到当前发版本的全量文件,非常方便,并且,大大减少了上传数据量,降低了网络耗时,大大提高了文件更新效率。

附图说明

[0029] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本申请的实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据提供的附图获得其他的附图。

[0030] 图1为本申请实施例提供的一种信息更新系统的结构图;

- [0031] 图2为本申请实施例提供了一种信息更新方法的时序图；
- [0032] 图3为本申请实施例提供了一种目录架构图；
- [0033] 图4为本申请实施例提供了一种RBU业务的目录结构输出界面；
- [0034] 图5为本申请实施例提供的另一种信息更新方法的时序图；
- [0035] 图6为本申请实施例提供了一种发布界面示意图；
- [0036] 图7为本申请实施例提供的又一种信息更新方法的流程图；
- [0037] 图8为本申请实施例提供了一种信息更新装置的结构框图；
- [0038] 图9为本申请实施例提供的另一种信息更新装置的结构框图；
- [0039] 图10为本申请实施例提供的又一种信息更新装置的结构框图；
- [0040] 图11为本申请实施例提供了一种服务器的硬件结构图。

具体实施方式

[0041] 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,都属于本申请保护的范围。

[0042] 为了使本申请的上述目的、特征和优点能够更加明显易懂,下面结合附图和具体实施方式对本申请作进一步详细的说明。

[0043] 图1为本申请实施例提供的信息更新系统的结构框图,该图所示的信息更新系统可以用于实现本申请实施例提供的信息更新方法。参照图1,该系统可以包括:服务器10以及至少一个客户端20;

[0044] 其中,服务器10为网络侧为用户提供服务的服务设备,其可能是多台服务器组成的服务器集群,也可能是单台服务器,在本实施例中,主要用来实现驱动程序和应用程序等产品新版本的发布,关于产品新版本发布过程可以参见下文相应实施例的描述。

[0045] 客户端20可以是装载在手机、平板电脑、笔记本电脑等用户设备,或者是其他工业设备,如工控机等设备上。在本申请实际应用中,通过与服务器10建立通信连接,用户可以通过客户端20访问服务器10,实现客户端20的各种功能。

[0046] 可选的,客户端20可以是安装在如上述所述设备上的应用程序,产品新版本开发人员完成新版本编译测试后,可以通过该客户端20将待发布的发布包上传到服务器10,以使服务器10按照以下实施例描述的方法,实现该发布包的发布。

[0047] 其中,当需要对终端上的应用程序、驱动程序或者是操作系统等产品进行升级更新时,可以通过互联网从服务器10下载相应的新版本信息,完成升级,从而实现对原版本产品功能的改进以及漏洞的修补。

[0048] 在实际应用中,本申请提供的信息更新方案可以在信息发布系统中实现,其中,信息发布系统通过由服务器、网络、终端等组成,对于应用开发技术人员,可以通过登录该信息发布系统,发布应用新版本或维护原版本应用的工作等等,由此可见,该信息发布系统可以认为是一个应用平台,本申请提供的信息更新方法可以在该应用平台中实现,且本申请提供的信息更新系统也可以适用于该应用平台,本申请对该应用平台的具体组成结构不作限定。

[0049] 其中,应用平台中通常会包括互联网数据中心,简称IDC,为互联网内容提供商(ICP)、企业、媒体和各类网站提供大规模、高质量、安全可靠的专业化服务器托管、空间租用、网络批发带宽以及ASP、EC等业务。本申请可以将其作为模块的部署地。

[0050] 可见,在开发人员进行软件开发时,通常是对应用平台的互联网数据中心中的相应应用功能模块进行开发和维护,且每个应用功能模块通常会分给不同的开发人员负责,以便提高整个应用的开发和维护效率。

[0051] 基于图1所示的信息更新系统,如图2所示,为本申请实施例提供的一种文件更新方法的时序图,该方法可以包括以下步骤:

[0052] 步骤S21,客户端向服务器发送针对目标业务的版本更新请求,该版本更新请求可以用于更新预设路径下存储的目标业务上次成功发布版本的全量文件。

[0053] 在本申请中,为了方便产品的每一个开发人员能够随时更新自己负责的功能部分,提高发布灵活性,可以通过增量发布的方式去更新目标业务中自己负责的功能部分,降低了发布包上传的网络耗时,从而提高了发布效率。

[0054] 基于此,在实际应用中,当开发人员需要对其负责的功能部分进行更新,编译并测试得到所需的发布包后,可以向服务器发起版本更新请求,本申请对该版本发布请求的发起方式不作限定,如通过触发客户端显示界面上针对当前发布版本的相应按钮,生成相应的版本更新请求并通过互联网发送至服务器等,本实施在此不再一一详述。

[0055] 当然,开发人员需要发布目标业务的初始发布版本时,也可以向服务器发起版本更新请求,本申请对该版本更新请求的触发条件不作限定,请求发布的当前发布版本可以是初始发布版本,也可以是升级发布版本。

[0056] 步骤S22,客户端接收到服务器反馈的响应消息,按照预设路径向服务器发送针对目标业务的当前发布版本的发布包;

[0057] 在实际应用中,通常会为产品的每个业务分配一个固定的上传路径即预设路径,并在该预设路径下存储相应业务的目录结构和发布包,这样,在开发人员针对某业务开发出初始版本或升级版本的发布包后,可以按照为其分配的预设路径将该发布包上传到服务器。

[0058] 例如,服务器可以给定该RBU业务的发布包的预设路径,如/data/release_pkg/BSI98747124/BSI98732084/BSI98750294/,但并不局限于此。此时,服务器检测到客户端发送的该RBU业务的当前发布版本的发布包,可以按照该预设路径,将发布包导入对应的位置,并在预设路径下生成该RBU业务的目录架构。

[0059] 另外,在本申请中,目标业务的当前发布版本的发布包可以单个文件、或者是由多个文件构成的文件夹压缩而成的压缩文件等,本申请对单个文件的具体文件格式、文件夹包含的各文件的文件格式,以及该文件夹压缩而成的压缩文件格式均不作限定,可以根据实际情况确定,但在本实施例中不同类型的发布包采用的发布方式是不同的。

[0060] 步骤S23,服务器检测该发布包的文件类型;

[0061] 可选的,本申请可以将单个文件(如.so、.a等格式的文件)构成的普通文件,以及由多个文件构成的文件夹并压缩成以.tgz.replace结尾的压缩文件即第一压缩文件划分为第一类文件;将其他格式的由包含多个文件的文件夹压缩而成的压缩文件即第二压缩文件划分为第二类文件。在本实施例中,可以预先设定第一类文件通过增量发布方式进行发

布,第二类文件通过全量发布方式进行发布。

[0062] 可选的,本实施例可以预先建立发布包的文件类型与发布方式之间的对应关系,这样,在检测到发布包的文件类型后,即可按照该对应关系确定当前发布版本的发布包的发布方式。

[0063] 需要说明的是,本申请对该对应关系的表示方式以及存储方式都不作限定,可以通过对应关系表的方式,也可以通过设置唯一标识,来确定发布包的文件类型与对应发布方式之间的对应关系等等,本申请在此不再一一列举。

[0064] 作为本申请另一实施例,对于开发人员来说,其上传的当前发布版本的发布包包含的文件类型是已知的,所以,本申请还可以在开发人员上传当前发布版本的发布包时,携带上传表征该发布包的发布方式的标识,从而使服务器根据接收到的标识,确定当前发布版本的发布包的发布方式,即全量发布或增量发布。

[0065] 由此可见,本申请对如何确定发布包的发布方式的过程不作限定,至于本次版本发布采用哪种发布方式与发布包的文件格式相关,具体如上相应部分的描述。

[0066] 步骤S24,服务器确定对该发布包进行增量发布,对目标业务上次成功发布版本的全量文件进行备份;

[0067] 在实际应用中,对于产品的开发人员来说,该产品的各业务的目录架构是清楚的,且对于不同的业务,其目录架构的一级目录通常是相同,如图3所示,一级目录可以包括real_body、body、rub以及release等文件夹。但该目录架构的二级及以下目录通常会因业务的不同而有所差别,具体可以根据业务的功能来确定,本实施例在此不再一一详述。

[0068] 需要说明的是,关于一级目录中各文件夹的名称并不局限于图3中的记载,可以根据文件夹的功能进行其他命名,并且,在图3所示的目录架构的架构中,二级及其以下目录如bin、conf、op、l4rank、mcp等可以表示业务文件本身包含的文件或文件夹,图3仅以RBU业务文件为例进行说明,对于其他业务文件,二级及其以下目录内容可能会有所不同,可根据具体情况确定,本申请在此不再一一详述。

[0069] 为了更清楚了解图3所示的目录架构,下面将对图3中各目录的文件夹存储信息的类型或者是各目录文件夹的功能进行说明,尤其是各业务共同具有的文件夹的功能。

[0070] 具体的,real_body文件夹可以用于保存上一次成功发布版本的全量文件;body文件夹可以用于临时保存当前发布版本的全量文件,当本次发布成功后,将body文件夹下的内容替换real_body文件夹下的内容,即利用当前发布版本的全量文件替换上一次发布版本的全量文件;rbu文件夹(其对应于RBU业务,对于其他业务,此文件夹可以是相应业务的文件夹)可以用于存放目标业务当前发布版本的发布包的目录结构;release文件夹可以用于保存业务文件各发布版本的发布包的全量文件和增量文件,v1和v2可以代表两个不同发布版本,changed文件夹可以用来保存增量文件,total文件夹可以用来保存全量文件。

[0071] 在实际应用中,对应于该目录架构中各文件夹的功能,可以将获得的信息导入对应的文件夹。在本实施例中,接收到RBU业务的当前发布版本的发布包后,通过上述方式确定对其进行增量发布,这种情况下,可以认为该业务已经存在一种发布版本的目录架构,即为上一次成功发布版本的全量文件的目录架构。

[0072] 可见,本申请在进行增量发布时,将要更新的目标业务的目录架构与上一次全量发布得到的目录架构完全一致。因此,步骤S14获得的目标业务的当前目录架构可以是该目

标业务上一次全量发布时生成的目录架构。

[0073] 结合上述对产品的各业务文件的目录架构的功能描述,如果直接在目录架构的一级目录real_body文件夹中进行全量文件的合并,那么,在发布到线上机器后,即进行灰度发布后,发现开发人员上传的发布包是错误的,此时,上传的错误文件已经与全量文件合并,将无法实现上一版本的回退,从而导致发布失败。

[0074] 为了解决发布包上传错误带来的问题,本实施例提出在确定对当前发布版本的发布包采用增量发布后,对当前目录架构存储的上一次成功发布版本的全量文件进行备份,即通过备份real_body文件夹中的全量文件的方式,使得服务器具有回退功能,这样,当开发人员或发布人员发现上传的发布包错误,可以重新获取real_body文件夹中上一次成功发布版本的全量文件,再进行增量发布,不会影响目标业务上一次成功发布版本的全量文件的完整性。

[0075] 基于此,服务器确定对当前发布版本的发布包进行增量发布,可以将目标业务所在目录架构中,用于临时存储当前发布版本的全量文件的文件夹确定为备份文件夹,并删除该备份文件夹中的内容,从而将用于存储上次成功发布版本的全量文件的文件夹内容拷贝至该备份文件夹。

[0076] 具体的,参照上图3所示的目录架构,服务器确定对当前发布版本的发布包进行增量发布,可以将body文件夹作为备份文件夹,删除body文件夹下的所有内容,再将real_body文件夹中的内容拷贝到body文件夹中,使得body/下的目录内容与real_body/下的目录内容一致,这样,在增量发布过程中,可以对body/下的全量文件进行合并处理,而real_body/下的全量文件仍然是上一次成功发布版本的全量发布包,即便本次上传的文件错误,仍可以采用上述方式重新拷贝real_body文件夹中的内容进行增量发布,保证了发布成功的可靠性。

[0077] 步骤S25,服务器检测当前发布版本的发布包,确定该发布包的类型;

[0078] 其中,结合上述描述可知,采用增量发布方式的发布包可以包括单个文件、第一压缩文件。在本申请中,若本次增量发布的目的是更新当前存储的某一个文件,可以直接采用文件替换的方式实现,参照图4,当需要对exam.so文件进行更新,开发人员可以重新开发exam.so文件,并用其替换原来的exam.so文件,这种情况下,开发人员上传的发布包通常是单个文件。

[0079] 然而,若本次发布版本的发布包包含的文件是上一次发布版本不存在的,且上一次发布版本的某文件在本次发布版本中不再使用,即成为废弃文件,对于这种情况,由于这两个文件不同,无法再使用文件替换的方式实现增量发布,本申请将采用文件所在的文件夹替换的方式实现增量发布,也就是说,这种情况下开发人员上传的发布包将会是第一压缩文件。

[0080] 由此可见,对于不同类型的增量发布包,具体采用的增量发布方式是不同的,本实施例中,服务器确定对接收到的当前发布版本的发布包采用增量发布方式后,还需要进一步确定该发布包的类型,即是单个文件还是特定格式压缩文件。

[0081] 步骤S26,服务器确定当前发布版本的发布包是单个文件,确定备份文件夹中的待更新文件,并利用该发布包替换待更新文件,进入步骤S211;

[0082] 结合上述分析,服务器确定本次上传的发布包需要进行增量发布后,通常会将目

标业务的目录结构发送至客户端输出,如图4所示,方便开发人员根据自己实际需要更新的功能部分,灵活选择本次发布包的应该上传的路径,并将该发布包的上传路径反馈至服务器,以使服务器按照该上传路径将接收到的发布包发送至目标业务的目录结构的相应位置。

[0083] 基于此,当开发人员本次上传发布的是单个文件,可以直接确定该文件在当前目录结构中的存储位置,以使服务器根据该存储位置或形成的上传路径,确定备份文件中需要替换的待更新文件,并利用接收到的单个文件替换该待更新文件,过程简单且方便,并不受该目标业务的其他开发人员的功能修改影响,大大提高了文件发布的灵活性以及效率。

[0084] 举例说明,结合上图4所示的一种业务文件的目录结构,若开发人员需要上传的是exam.so格式的单个文件,开发人员可以根据客户端输出的RBU业务文件的目录架结构,确定该发布包应该上传到该目标业务的目录结构什么位置,即确定增量发布包的上传路径,以使服务器按照该上传路径将增量发布包导入该RBU业务文件的目录结构的对应位置,并完成单个文件的增量发布。

[0085] 需要说明的是,本申请对确定单个文件的上传路径方式不作限定,相应的,对确定备份文件中的待更新文件的方式不作限定,可以采用如上述描述,由开发人员直接从客户端显示的目标业务的目录结构中选择确定,也可以由开发人员直接输入单个文件的上传路径等,本申请在此不再一一详述。

[0086] 步骤S27,服务器确定当前发布版本的发布包是第一压缩文件,创建存储文件夹,并将该第一压缩文件导入存储文件夹进行解压,得到第一解压文件夹;

[0087] 如上述分析,在增量发布中,对于上一次成功发布版本中的某个文件,若本次发布版本中不再需要该文件,该文件作为废弃文件,通常需要将其删除,以保证本次发布版本成功发布后的业务能够正常工作,这种情况下,本实施例将采用文件夹替换的方式,即直接确定需要删除的废弃文件的上一级目录文件夹,之后,直接利用本次发布版本中的该文件夹进行替换。

[0088] 举例说明,参照上图3所示,若在conf/文件夹下,在v1发布版本时,有a.conf文件,当在v2发布版本时,不再需要a.conf文件,而是替换成b.conf文件,此时就需要将a.conf文件删除,只保留b.conf文件。对于这种情况,本申请可以将v2发布版本的conf/文件夹整体替换v1发布版本时的conf/文件夹。由此可见,本申请将文件的更新上升为文件夹的替换,来解决废弃文件删除的问题。

[0089] 在实际应用中,在进行文件夹发布过程中,通常会采用压缩文件方式进行传输,但对文件夹的具体压缩方式不作限定,本实施例对于需要采用增量发布方式进行发布的压缩文件,可以是.tgz.replace格式的第一压缩文件。

[0090] 在本实施例中,服务器确定接收到的是第一压缩文件时,可以创建一个新的目录文件夹即存储文件夹,来完成压缩文件的解压工作。如上图3所示,可以在目标业务的预设路径下的目录架构中,创建tmp文件夹,来暂时存放本次上传的发布包,并解压得到该发布包包含的文件夹,即.tgz.replace格式的文件夹。在服务器确定本次发布版本成功发布后,该发布包及其解压文件将不再需要,可以直接删除该存储文件夹如tmp文件夹,简化所得目录架构,节省存储空间。

[0091] 步骤S28,服务器在备份文件夹以及业务文件夹中,确定与该第一解压文件夹名称

相同且路径级别相同的文件夹为待更新文件夹；

[0092] 仍以上举例的conf文件夹的替换为例进行说明,在创建的存储文件夹中对发布包进行解压,得到conf文件夹后,结合上图3所示的当前目录架构,可以确定body/(即备份文件夹)以及rbu/(即RBU业务的业务文件夹,对于其他业务,该业务文件夹名称可以相应改变目录下同级路径的文件夹中,与解压得到的conf文件夹属于同级目录的文件夹,如将body/conf和rbu/conf确定为待更新文件夹。

[0093] 其中,业务文件夹如上图3所示的rbu/文件夹中,包含的可以是该发布包的目录结构以及发布包(即增量发布包),所以,当需要开发人员确定发布包的具体上传路径时,可以将该rbu文件夹中的目录结构反馈至客户端输出,如上图4所示。

[0094] 步骤S29,服务器将待更新文件夹替换成第一解压文件夹,获得当前发布版本的全量文件和增量文件；

[0095] 继上文描述,可以利用第一解压文件夹,如conf文件夹替换body/目录下的conf文件夹及rbu/目录下的conf文件夹,从而使body/目录下重新形成针对当前发布版本的全量文件,即更新了备份文件夹中存储的上一次成功发布版本的全量文件,同时也更新了发布包文件夹下存储的增量文件,得到rbu/目录下重新形成当前发布版本的增量文件。

[0096] 步骤S210,服务器在目标业务所在目录架构的版本文件夹中,创建当前发布版本的增量文件夹和全量文件夹,并将得到的全量文件和增量文件导入相应文件夹；

[0097] 结合上述图3对目标业务所在的目录架构的描述,对于该目标业务同一发布版本的全量文件和增量文件,可以存放至同一版本文件夹下的两个不同文件夹中,例如,生成release/版本号/changed和release/版本号/total,之后,将增量文件导入changed文件夹存储,将全量文件导入total文件夹存储,以便后续按照版本路径获取所需文件。

[0098] 步骤S211,服务器将当前目录架构中,存储上次成功发布版本的全量文件的文件夹的内容替换成当前备份文件夹中的内容。

[0099] 在本实施例中,确定本次版本发布成功后,可以将body文件夹即备份文件夹中的内容移动到real_body文件夹(即存储上次成功发布版本的全量文件的文件夹)中,从而使real_body文件夹存储当前发布版本的全量文件,完成本次发布包的增量发布。

[0100] 综上所述,在本实施例中,当开发人员需要对目标业务自己负责的功能部分进行升级,并确定相应的发布包后,可以采用增量发布方式实现该发布包的发布,无需上传和发布整个全量文件,且不需要与该目标业务的其他开发人员进行配合,就可实现当前发布版本的发布简单且方便,且节省了开发人员上传发布包的时间以及运维人员发布时间,提高了产品新版本发布效率。

[0101] 如图5所示,为本申请实施例提供的另一种文件更新方法的时序图,本实施例主要是对新版本产品文件的增量发布方式进行说明,具体可以包括以下步骤:

[0102] 步骤S51,客户端向服务器发送针对目标业务的版本更新请求,该版本更新请求可以用于更新预设路径下存储的目标业务上次成功发布版本的全量文件；

[0103] 可选的,在本实施例中,服务器可以对发送该版本更新请求的客户端进行身份验证,如验证客户端的登录信息或注册信息等,判断该客户端用户是否具有目标业务新版本的发布权限。需要说明的是,本申请对客户端的身份验证方式不作限定。

[0104] 步骤S52,客户端接收到服务器反馈的响应消息,输出版本发布界面；

[0105] 在本实施例中,服务器验证该客户端身份合格后,可以向该客户端反馈相应的响应消息,之后,客户端可以输出版本发布界面,如图6所示,该版本发布界面上可以设置版本发布方式的选择按钮,开发人员可以选择目标业务的当前发布版本的发布包的发布方式,但并不局限于图6所示的界面内容。

[0106] 步骤S53,服务器接收到客户端反馈的增量发布指令,对目标业务上次成功发布版本的全量文件进行备份,并将该目标业务的当前目录结构发送至客户端;

[0107] 在本实施例中,客户端输出如图6所示的版本发布界面时,开发人员可以根据实际发布需求,选择本次发布包的发布方式,如选定增量发布方式时,客户端将生成相应的增量发布指令并发送至服务器,由于需要开发人员确定本次发布包的具体上传路径,所以,服务器可以将该目标业务的当前目录结构,即上一次成功发布的全量文件的目录结构发送至客户端。

[0108] 其中,关于对目标业务上次成功发布版本的全量文件的备份过程可以参照上述实施例相应部分的描述,本实施例在此不再赘述。

[0109] 步骤S54,客户端输出目标业务的当前目录结构,获得当前发布版本的发布包的上传路径,并按照该上传路径将当前发布版本的发布包发送至服务器;

[0110] 在本实施例中,客户端输出的目录业务的当前目录结构可以如图4所示,但并不局限于此,此时,开发人员可以直接从该目录结构中确定需要替换的待更新文件,从而确定本次发布包的上传路径,之后,客户端就能够按照该上传路径,将当前发布版本的发布包上传至服务器存储的该目标业务的目录架构中的业务文件夹。

[0111] 步骤S55,服务器检测该发布包的类型;

[0112] 如上述实施例的描述,在增量发布过程中,对于单个文件和文件夹的发布过程并不相同,所以,服务器在接收到增量发布包后,需要先对该发布包包含的文件类型进行检测,判断其包含的是单个文件,还是由文件夹压缩而成的压缩文件。

[0113] 步骤S56,服务器确定当前发布版本的发布包是单个文件,利用该发布包替换备份文件夹中的待更新文件;

[0114] 对于上传的单个文件,本实施例可以采用直接替换的方式,即从备份文件夹中,查找到上一次发布版本中与该文件同名的待更新文件,并将该待更新文件直接替换为该文件。

[0115] 步骤S57,服务器确定当前发布版本的发布包是第一压缩文件,在目标业务所在的目录架构中创建存储文件夹,并将该第一压缩文件导入存储文件夹进行解压,得到第一解压文件夹;

[0116] 其中,此处的第一压缩文件可以由多个文件构成的文件夹压缩而成,且以.tgz.replace格式结尾的压缩文件。

[0117] 步骤S58,服务器在备份文件夹以及业务文件夹中,确定与第一解压文件夹名称相同且路径级别相同的文件夹为待更新文件夹;

[0118] 步骤S59,服务器将待更新文件夹替换成第一解压文件夹,获得当前发布版本的全量文件和增量文件;

[0119] 步骤S510,服务器在当前目录架构的版本文件夹中,创建对应版本的增量文件夹和全量文件夹,并将得到的全量文件和增量文件导入相应文件夹;

[0120] 步骤S511,服务器将当前目录架构中,存储上次成功发布版本的全量文件的文件夹的内容替换成当前备份文件夹中的内容。

[0121] 综上,在本实施例中,开发人员需要对目标业务文件中的单个文件或文件夹进行更新时,可以将其作为发布包,并利用客户端输出的该目标业务的目录结构确定其目标上传路径,将该发布包上传到服务器的相应位置,从而使服务器采用相应的逻辑实现该发布包的增量发布,与上传全量文件进行发布包发布的方式相比,避免了对与上次发布版本相同文件的上传,大大减少了上传文件量,节省了开发人员上传发布包的时间以及网络流量,提高了产品新版本发布效率。

[0122] 而且,本申请采用增量发布实现新版本发布包的发布,开发人员不需要得知其他开发人员负责的功能部分,能够将自己负责的功能部分直接上传,服务器就能够实现这部分功能的发布,发布过程比较灵活且实用。

[0123] 下面将对上述实施例描述的当前发布版本的发布包的全量发布方式的实现过程进行说明,参照图7所示,本实施例主要是从服务器角度描述全量发布过程,关于客户端向服务器发起的版本更新请求过程可以参照上述实施例的描述,本实施例在此不再详述,因此,全量发布过程可以包括:

[0124] 步骤S71,接收客户端上传的针对目标业务的当前发布版本的发布包;

[0125] 在实际应用中,服务器会为产品的每一个业务文件分配固定的上传路径,以使该业务文件的所有发布包都上传到该上传路径下。因此,在本实施例中,服务器确定开发人员本次上传的发布包需要进行全量发布后,可以将该目标业务对应的预设路径即上传路径发送至客户端,以使客户端按照该预设路径上传当前发布版本的发布包。

[0126] 步骤S72,对该发布包进行解压,并利用得到的第二解压文件夹生成rub文件夹下的目录结构;

[0127] 结合上述描述,对于全量发布包,通常是由多个文件构成的文件夹压缩而成的第二压缩文件,且该第二压缩文件不是以.tgz.replace格式结尾的压缩文件。所以,在接收到增量发布包后,通常需要先对其进行解压,并生成发布包文件夹下的目录结构,即如图3中的rbu/文件夹下的目录结构,

[0128] 需要说明的是,本实施例描述的全量发布过程仅以图3所示的RBU业务的目录结构为例进行说,对于其他业务,图3的一级目录文件夹不变,但二级及其以下目录文件夹组成将会相应改变,本实施例在此不再一一详述。

[0129] 步骤S73,创建body文件夹及其目录结构,并将第二解压文件夹导入body文件夹的目录结构下;

[0130] 步骤S74,创建release文件夹及其目录结构,并将全量解压文件导入该release文件夹中的相应版本号的全量文件夹;

[0131] 如图3所示,release文件夹下可以分成多个版本号,每个版本号下可以分为全量文件夹和增量文件夹,对于本次全量发布包,可以相应的版本号后,直接将解压文件即全量文件导入该版本号下的全量文件夹。

[0132] 步骤S75,将rbu文件夹中的非文件夹的文件删除,获得目标业务的当前发布版本的目录结构;

[0133] 步骤S76,将当前body文件夹中的内容移动至real_body文件夹中。

[0134] 在本实施例中,通过上述全量发布方式实现当前发布版本的发布包的发布后,可以在其得到的目录架构的基础上,开发人员根据实际需要进行增量发布,实现该业务的新版本的发布,过程可以参照上述实施例描述的过程,本实施例在此不再详述。

[0135] 需要说明的是,在实际应用中,通常是对某业务的初始版本发布时采用上述全量发布方式,而对于在该初始版本的基础上进行改进,得到的新版本的发布,通常是采用上述实施例描述的增量发布,但并不局限于。

[0136] 由此可见,在对目标业务的当前发布版本的发布包的发布时,可以根据实际情况,灵活选择采用全量发布方式或增量发布方式,保证当前发布版本的发布包的可靠发布。

[0137] 而且,在服务器完成全量发布后,可以将生成的针对该发布版本的目录结构反馈至客户端,由客户端输出该目录结构,以便开发人员能够直观得知目标业务当前发布版本的目录结构,此时,通过点击还可以查看各级目录各文件包含的内容。另外,开发人员还可以在此基础上进行增量发布,过程可以参照上文相应实施例的描述。

[0138] 如图8所示,为本申请实施例提供的一种信息更新装置的结构框图,该信息更新装置可以包括:

[0139] 数据传输模块81,用于接收客户端上传的针对目标业务的当前发布版本的发布包;

[0140] 第一数据处理模块82,用于解析所述发布包的文件类型,确定对所述发布包进行增量发布,触发所述数据传输模块将所述目标业务的当前目录结构反馈至所述客户端输出,以使所述客户端基于所述当前目录结构确定所述发布包的目标上传路径;

[0141] 第一数据导入模块83,用于按照所述目标上传路径,将所述发布包导入所述当前目录结构的对应文件夹;

[0142] 文件更新模块84,用于利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,得到当前发布版本的全量文件。

[0143] 可选的,在上述实施例的基础上,为了实现对当前发布版本的发布包的增量发布,如图9所示,信息更新装置还可以包括:

[0144] 文件备份模块85,用于对所述目标业务上次成功发布版本的全量文件进行备份;

[0145] 相应地,文件更新模块84具体可以用于利用导入的所述的发布包,更新得到的备份文件夹,所述备份文件夹包含在所述目标业务所在的目录架构中,并利用更新后的备份文件夹内容,替换所述目录架构中用于存储上次成功发布版本的全量文件的文件夹内容。

[0146] 其中,当发布包具体是第一压缩文件时,信息更新装置还可以包括:

[0147] 第一创建模块86,用于在所述目标业务所在的目录架构中创建存储文件夹;

[0148] 则文件更新模块84具体可以用于将导入的第一压缩文件发送至所述存储文件夹进行解压,得到相应的第一解压文件夹,查询所述目录架构下与所述第一解压文件夹名称相同,且路径级别相同的文件夹,并将查询到的文件夹确定为待更新文件夹,将所述待更新文件夹替换成所述第一解压文件夹,获得当前发布版本的全量文件。

[0149] 可选的,信息更新装置还可以包括:

[0150] 第一生成模块87,用于在将所述发布包导入所述当前目录结构的对应文件夹时,生成所述目标业务的增量文件;

[0151] 第二创建模块88,用于在所述目标业务所在目录架构的版本文件夹中,创建当前

发布版本对应全量文件夹和增量文件夹；

[0152] 第二数据导入模块89,用于将得到的所述当前发布版本的全量文件导入所述全量文件夹,并将得到的所述增量文件导入所述增量文件夹。

[0153] 需要说明的是,关于本申请提供的上述各服务器实施例实现增量发布的具体过程可以参照上述方法实施例相应部分的描述,本实施例在此不再赘述。

[0154] 作为本申请又一实施例,根据实际需要,服务器还可以对目标业务的当前发布版本的发布包进行全量发布,如图10所示,该信息更新装置还可以包括:

[0155] 第二数据处理模块810,用于当所述发布包是第二压缩文件,确定对所述发布包进行全量发布;

[0156] 解压模块811,用于对所述第二压缩文件进行解压;

[0157] 第二生成模块812,用于利用解压得到的文件或文件夹,生成所述目标业务的目录结构,以及用于存储当前发布版本发布包的文件夹的目录结构;

[0158] 第三数据导入模块813,用于将解压得到的文件或文件夹导入所述文件夹的目录结构中相应文件夹,获得当前发布版本的全量文件;

[0159] 第三创建模块814,用于在所述目标业务所在目录架构的版本文件夹中,创建当前发布版本对应全量文件夹,并将获得的全量文件导入所述全量文件夹。

[0160] 需要说明的是,关于信息更新装置利用上述功能模块实现发布包的全量发布的具体过程,可以参照上述方法实施例相应部分的描述,本实施例在此不再赘述。

[0161] 由此可见,本申请提供的信息更新装置可以根据客户端上传的发布包的类型,确定其发布方式是增量发布还是全量发布,从而根据相应的控制策略,实现对目标业务当前发布版本的发布包的发布,方便开发人员根据实际发布需要,灵活选择发布方式完成当前发布版本的发布包的发布,非常实用。尤其是仅对自己负责的部分功能进行升级,可以直接采用增量发布方式完成方式,不需要与其他开发人员进行配合,非常方便,大大提高了发布效率。

[0162] 下面将从服务器的硬件结构进行介绍,参照图11所示,为本申请实例提供的一种服务器的硬件结构图,该服务器可以包括:处理器111,通信接口112、存储器113以及通信总线114。

[0163] 其中,处理器111、通信接口112、存储器113通过通信总线114完成相互间的通信;

[0164] 可选的,通信接口112可以为通信模块的接口,如GSM模块的接口;

[0165] 处理器111可能是一个中央处理器CPU,或者是特定集成电路ASIC (Application Specific Integrated Circuit),或者是被配置成实施本申请实施例的一个或多个集成电路。

[0166] 存储器113可能包含高速RAM存储器,也可能还包括非易失性存储器 (non-volatile memory),例如至少一个磁盘存储器。本实施例中,该存储器113可以存储实现上述信息更新方法的多个指令。

[0167] 在本申请中,处理器111加载并执行存储器中的多个指令,具体用于:

[0168] 接收客户端上传的针对目标业务的当前发布版本的发布包;

[0169] 检测所述发布包的文件类型,确定对所述发布包进行增量发布;

[0170] 将所述目标业务的当前目录结构反馈至所述客户端输出,以使所述客户端基于所

述当前目录结构确定所述发布包的目标上传路径；

[0171] 按照所述目标上传路径,将所述发布包导入所述当前目录结构的对应文件夹；

[0172] 利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,得到当前发布版本的全量文件。

[0173] 需要说明的是,关于处理器111实现上述功能的具体过程可以参照上述方法实施例相应部分的描述,本实施例在此不再详述。

[0174] 结合上图1,本申请实施例还提供了一种信息更新系统,该系统可以包括服务器10以及至少一个客户端20,其中:

[0175] 客户端20,用于按照预设路径,向所述服务器上传针对目标业务的当前发布版本的发布包;输出所述服务器反馈的所述目标业务的当前目录结构,并基于所述当前目录结构,确定所述发布包的目标上传路径;

[0176] 服务器10,用于解析所述发布包的文件类型,确定对所述发布包进行增量发布,并将所述目标业务的当前目录结构反馈至所述客户端,按照所述客户端发送的所述目标上传路径,将所述发布包导入所述当前目录结构的对应文件夹,并利用导入的所述发布包,更新所述目标业务上次成功发布版本的全量文件,得到当前发布版本的全量文件。

[0177] 其中,关于服务器10以及客户端20在信息更新过程中的功能实现,可以参照上述方法实施例相应部分的描述,本实施例在此不再详述。

[0178] 综上,本申请提供的信息更新方案,能够根据发布需要,灵活选择全量发布或增量发布包,尤其是当开发人员需要对自己负责的功能模块进行升级,能够直接将其开发的发布包上传至服务器,不需要其他开发人员开发的其他功能模块的发布包的配合,服务器即可采用增量发布方式对接收到的部分功能的发布包的发布,完成这部分功能的更新,非常方便,且大大降低了上传的网络耗时,提高了发布效率。

[0179] 最后,需要说明的是,关于上述各实施例中,诸如第一、第二等之类的关系术语仅仅用来将一个操作、单元或模块与另一个操作、单元或模块区分开来,而不一定要求或者暗示这些单元、操作或模块之间存在任何这种实际的关系或者顺序。而且,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、方法或者系统不仅包括那些要素,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、方法或者系统所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法或者系统中还存在另外的相同要素。

[0180] 本说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似部分互相参见即可。对于实施例公开的服务器而言,由于其与实施例公开的方法对应,所以描述的比较简单,相关之处参见方法部分说明即可。

[0181] 专业人员还可以进一步意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。

[0182] 结合本文中所公开的实施例描述的方法或算法的步骤可以直接用硬件、处理器执行的软件模块,或者二者的结合来实施。软件模块可以置于随机存储器(RAM)、内存、只读存储器(ROM)、电可编程ROM、电可擦除可编程ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。

[0183] 对所公开的实施例的上述说明,使本领域专业技术人员能够实现或使用本申请。对这些实施例的多种修改对本领域的专业技术人员来说将是显而易见的,本文中所定义的一般原理可以在不脱离本申请的精神或范围的情况下,在其它实施例中实现。因此,本申请将不会被限制于本文所示的这些实施例,而是要符合与本文所公开的原理和新颖特点相一致的最宽的范围。

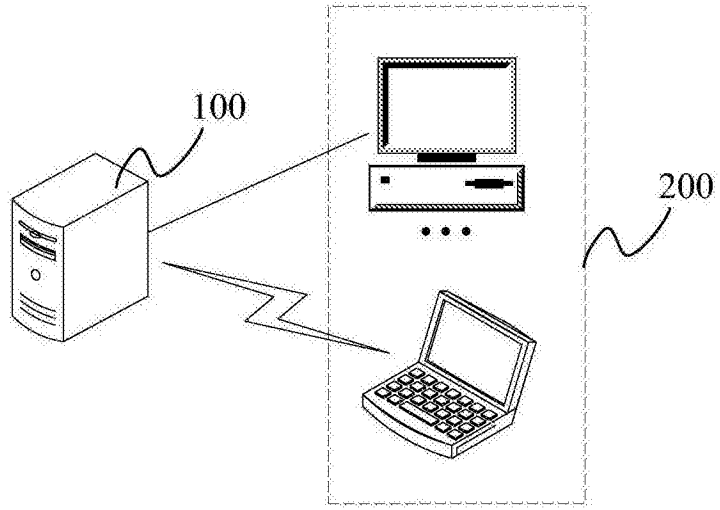


图1

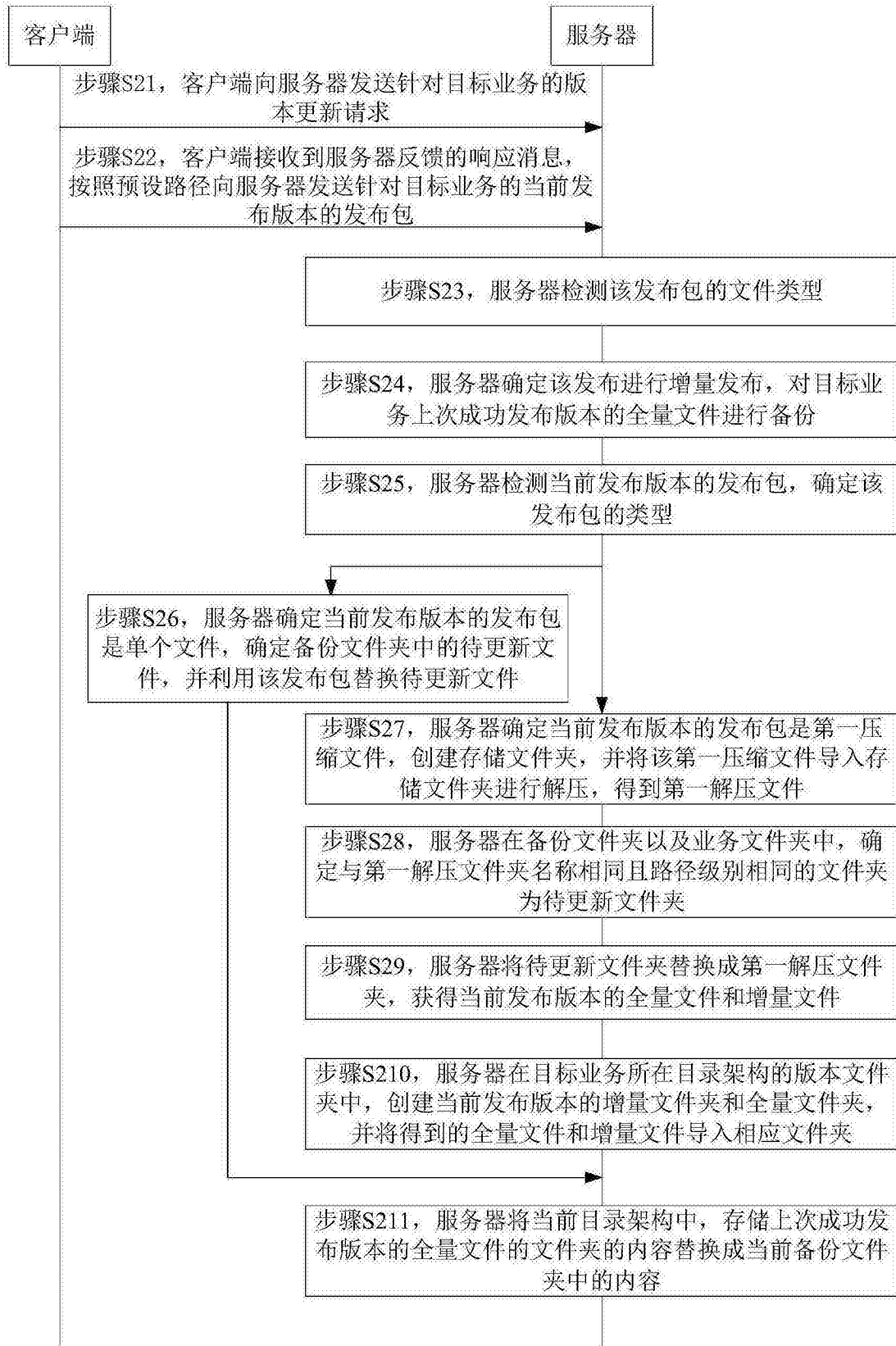


图2

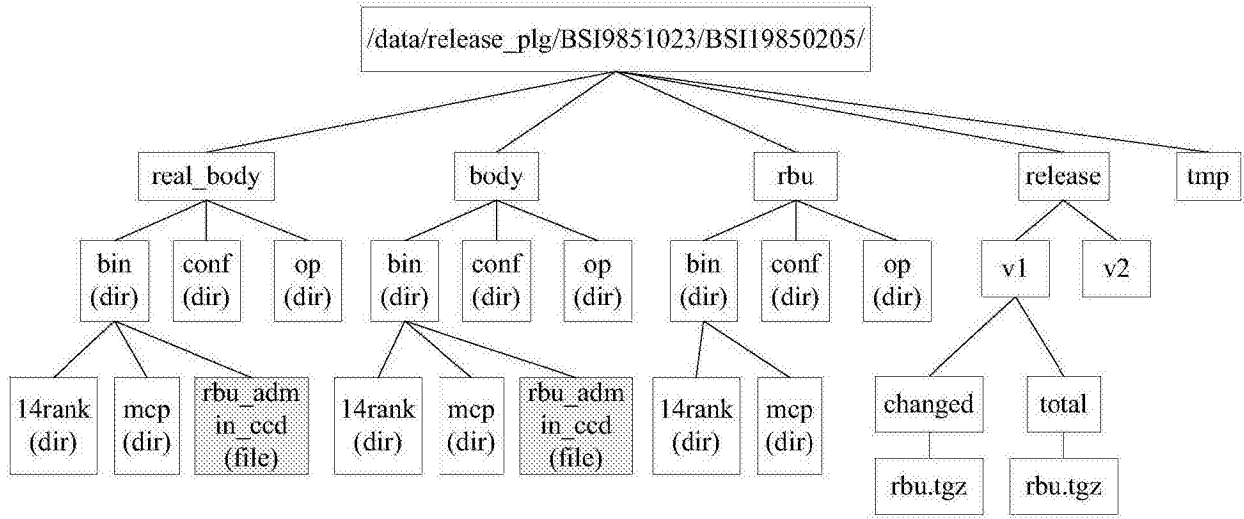


图3

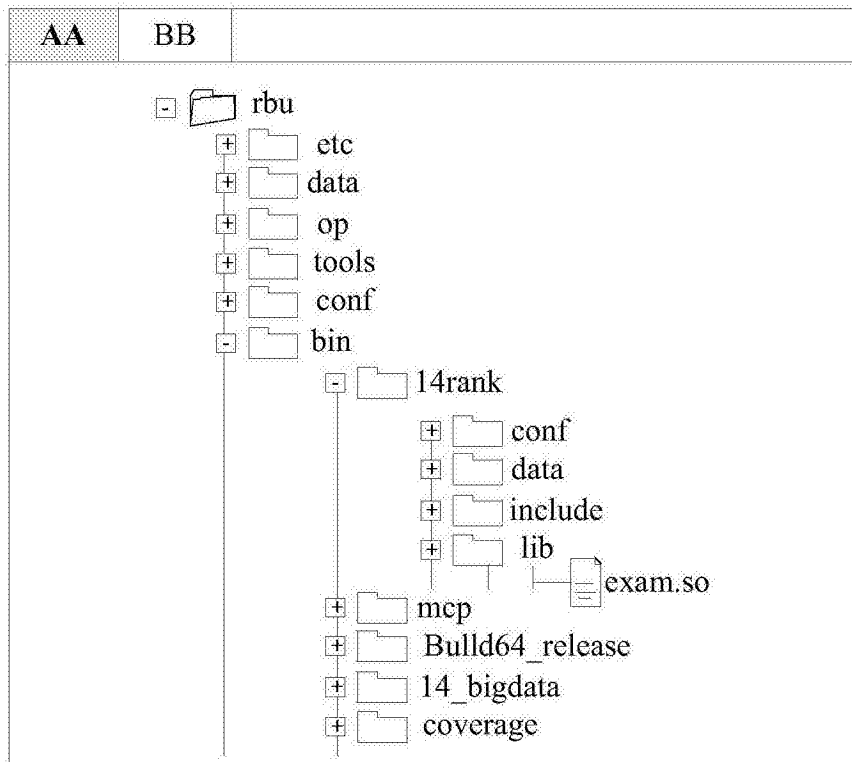


图4

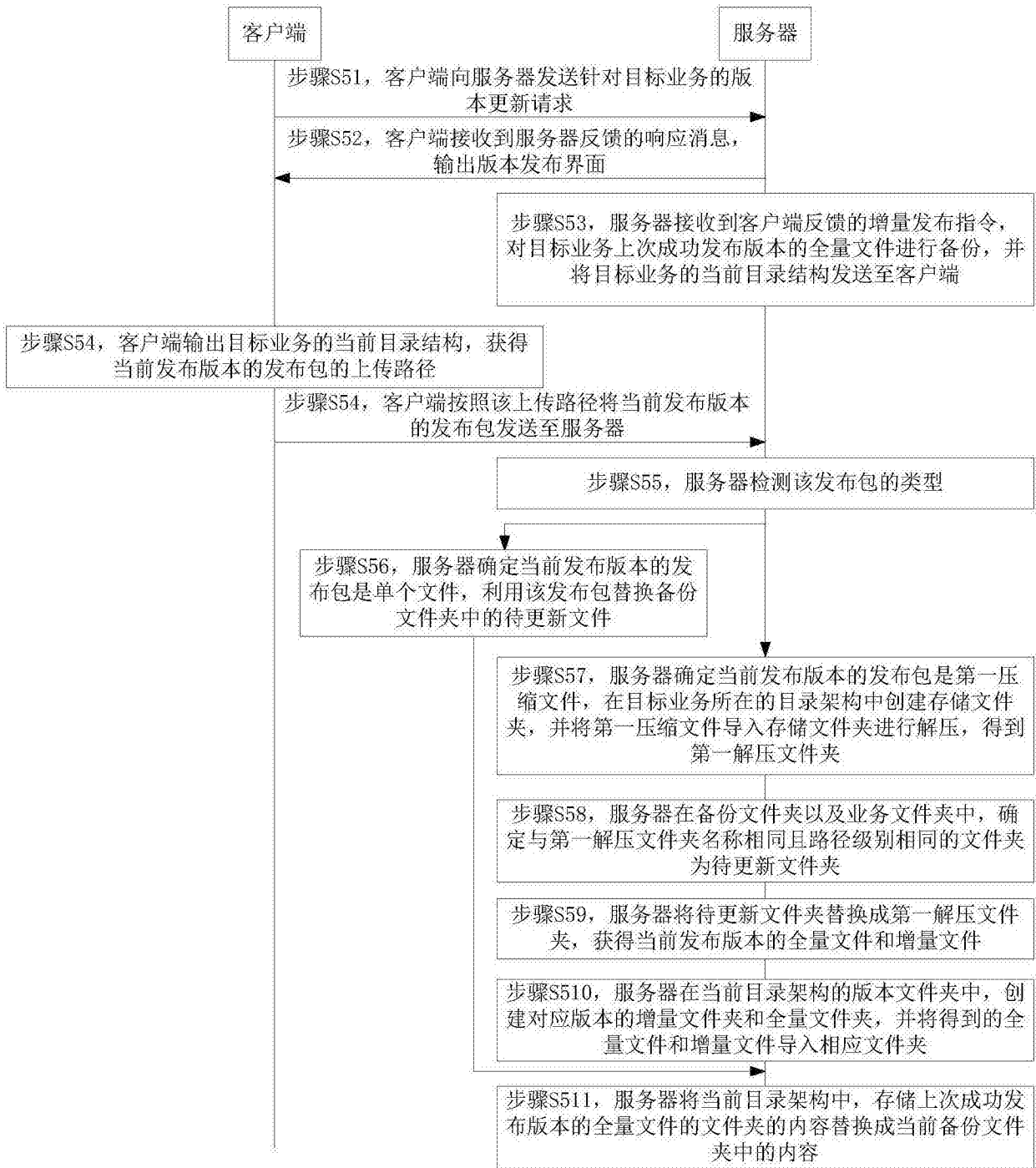


图5

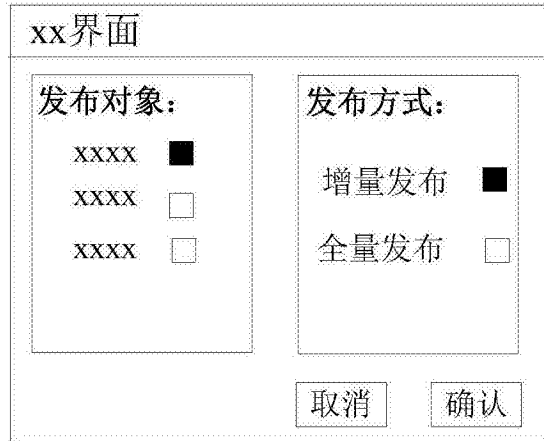


图6



图7



图8

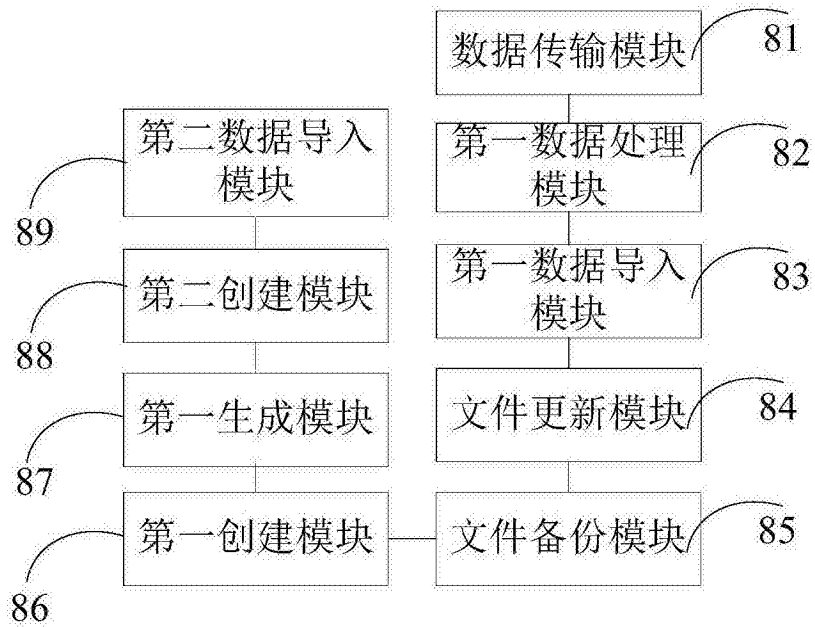


图9

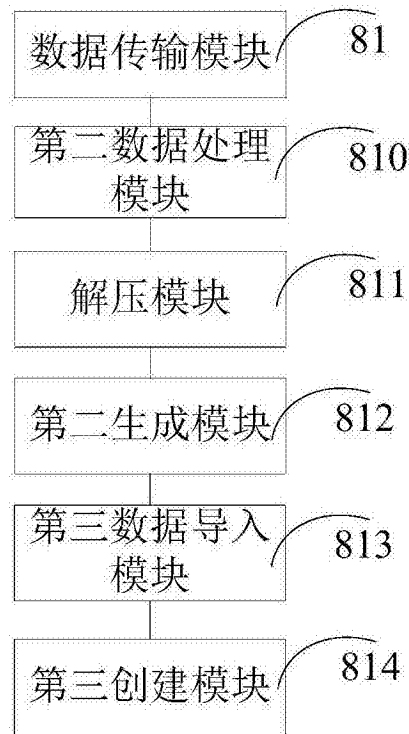


图10

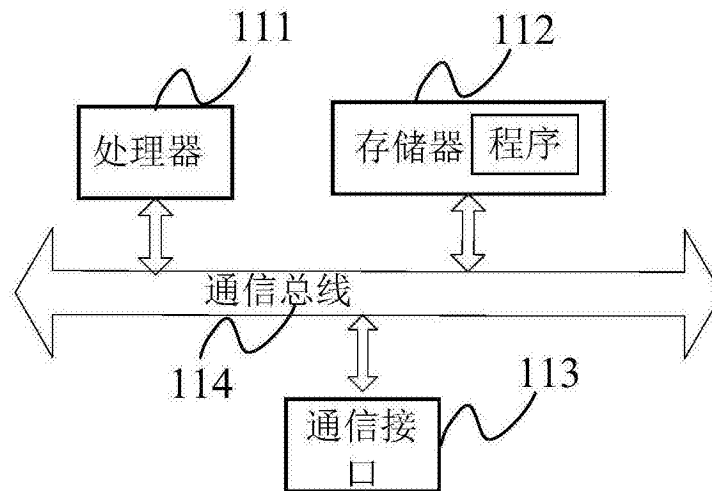


图11