



(12) 发明专利

(10) 授权公告号 CN 111522574 B

(45) 授权公告日 2024. 05. 03

(21) 申请号 202010144853.8

G06F 16/903 (2019.01)

(22) 申请日 2020.03.04

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 107885526 A, 2018.04.06

申请公布号 CN 111522574 A

CN 108256587 A, 2018.07.06

CN 110231947 A, 2019.09.13

(43) 申请公布日 2020.08.11

US 2004210551 A1, 2004.10.21

US 6425125 B1, 2002.07.23

(73) 专利权人 平安科技(深圳)有限公司

地址 518000 广东省深圳市福田区福田街
道福安社区益田路5033号平安金融中
心23楼

审查员 龙绘蓝

(72) 发明人 李桃

(74) 专利代理机构 深圳市赛恩倍吉知识产权代
理有限公司 44334

专利代理师 林丽纯

(51) Int. Cl.

G06F 8/658 (2018.01)

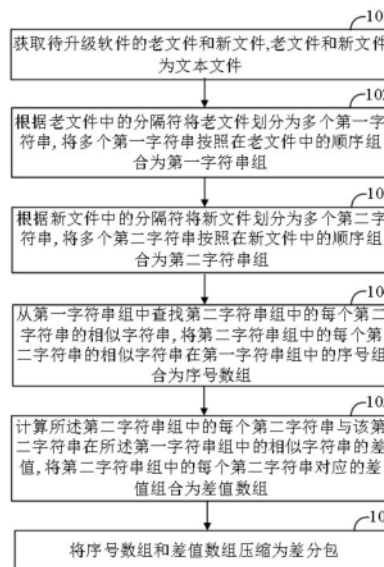
权利要求书3页 说明书14页 附图2页

(54) 发明名称

差分包生成方法及相关设备

(57) 摘要

本发明提供了一种差分包生成方法及相关设备。所述方法获取待升级软件的老文件和新文件;将所述老文件划分为多个第一字符串,将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组;将所述新文件划分为多个第二字符串,将所述多个第二字符串按照在所述新文件中的顺序组合为第二字符串组;将所述第二字符串组中的每个第二字符串的相似字符串在所述第一字符串组中的序号组合为序号数组;计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值,将所述第二字符串组中的每个第二字符串对应的差值组合为差值数组;将所述序号数组和所述差值数组压缩为差分包。本发明提升了生成差分包的效率。



1. 一种差分包生成方法,其特征在于,所述方法包括:

获取待升级软件的老文件和新文件,所述老文件和所述新文件为文本文件;

根据所述老文件中的分隔符将所述老文件划分为多个第一字符串,将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组;

根据所述新文件中的分隔符将所述新文件划分为多个第二字符串,将所述多个第二字符串按照在所述新文件中的顺序组合为第二字符串组;

从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串,将所述第二字符串组中的每个第二字符串的相似字符串在所述第一字符串组中的序号组合为序号数组;其中,所述从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串包括:对于所述第二字符串组中的任意一个第二字符串 A_i ,确定第二字符串 A_i 包含的字符的数量 m_i ,确定第二字符串 A_i 的各个字符 $a_{ix}, x=0,1,2,\dots,m_i-1$;

对于所述第一字符串组中的任意一个第一字符串 B_j ,确定第一字符串 B_j 包含的字符的数量 n_j ,确定第一字符串 B_j 的各个字符 $b_{jy}, y=0,1,2,\dots,n_j-1$;

计算第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的各个中间编辑距离 $v_{ij}(x+1, y+1), x=0,1,2,\dots,m_i-1, y=0,1,2,\dots,n_j-1$:

$$v_{ij}(x+1, y+1) = \min \begin{cases} v_{ij}(x+1, y) \\ v_{ij}(x, y+1) \\ v_{ij}(x, y) + \begin{cases} 1 & \text{if } a_{ix} \neq b_{jy} \\ 0 & \text{if } a_{ix} = b_{jy} \end{cases} \end{cases}$$

其中, $v_{ij}(x, 0) = x, v_{ij}(0, y) = y$,第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的编辑距离为 $v_{ij}(m_i, n_j)$;

从所述第一字符串组中查找与第二字符串 A_i 的编辑距离最小的第一字符串,将所述编辑距离最小的第一字符串作为第二字符串 A_i 的相似字符串;

计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值,将所述第二字符串组中的每个第二字符串对应的差值组合为差值数组;其中,所述计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值包括:对所述第二字符串组中的每个第二字符串的二进制值和该第二字符串在所述第一字符串组中的相似字符串的二进制值进行对齐处理,得到所述第二字符串组中的每个第二字符串的中间二进制值和该第二字符串在所述第一字符串组中的相似字符串的中间二进制值;

将所述序号数组和所述差值数组压缩为差分包。

2. 如权利要求1所述的方法,其特征在于,所述将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组包括:

删除所述多个第一字符串中重复的第一字符串,得到多个去重后的第一字符串;

将所述多个去重后的第一字符串按照在所述老文件中的顺序组合为所述第一字符串组。

3. 如权利要求1-2中任一项所述的方法,其特征在于,在所述获取待升级软件的老文件和新文件之前,所述方法还包括:

获取所述老文件的描述信息；

根据所述老文件的描述信息判断所述老文件是否合法。

4. 如权利要求1-2中任一项所述的方法,其特征在于,所述方法还包括:

响应于客户端的下载请求,将所述差分包发送给所述客户端,使所述客户端根据所述差分包和所述老文件生成所述新文件。

5. 如权利要求1-2中任一项所述的方法,其特征在于,所述方法还包括:

将所述差分包分发送给服务器。

6. 一种差分包生成装置,其特征在于,所述装置包括:

获取模块,用于获取待升级软件的老文件和新文件,所述老文件和所述新文件为文本文件;

第一划分模块,用于根据所述老文件中的分隔符将所述老文件划分为多个第一字符串,将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组;

第二划分模块,用于根据所述新文件中的分隔符将所述新文件划分为多个第二字符串,将所述多个第二字符串按照在所述新文件中的顺序组合为第二字符串组;

查找模块,用于从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串,将所述第二字符串组中的每个第二字符串的相似字符串在所述第一字符串组中的序号组合为序号数组;其中,所述从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串包括:对于所述第二字符串组中的任意一个第二字符串 A_i ,确定第二字符串 A_i 包含的字符的数量 m_i ,确定第二字符串 A_i 的各个字符 a_{ix} , $x=0,1,2,\dots,m_i-1$;

对于所述第一字符串组中的任意一个第一字符串 B_j ,确定第一字符串 B_j 包含的字符的数量 n_j ,确定第一字符串 B_j 的各个字符 b_{jy} , $y=0,1,2,\dots,n_j-1$;

计算第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的各个中间编辑距离 $v_{ij}(x+1,y+1)$, $x=0,1,2,\dots,m_i-1,y=0,1,2,\dots,n_j-1$:

$$v_{ij}(x+1,y+1) = \min \begin{cases} v_{ij}(x+1,y) \\ v_{ij}(x,y+1) \\ v_{ij}(x,y) + \begin{cases} 1 & \text{if } a_{ix} \neq b_{jy} \\ 0 & \text{if } a_{ix} = b_{jy} \end{cases} \end{cases}$$

其中, $v_{ij}(x,0)=x,v_{ij}(0,y)=y$,第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的编辑距离为 $v_{ij}(m_i,n_j)$;

从所述第一字符串组中查找与第二字符串 A_i 的编辑距离最小的第一字符串,将所述编辑距离最小的第一字符串作为第二字符串 A_i 的相似字符串;计算模块,用于计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值,将所述第二字符串组中的每个第二字符串对应的差值组合为差值数组;其中,所述计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值包括:对所述第二字符串组中的每个第二字符串的二进制值和该第二字符串在所述第一字符串组中的相似字符串的二进制值进行对齐处理,得到所述第二字符串组中的每个第二字符串的中间二进制值和该第二字符串在所述第一字符串组中的相似字符串的

中间二进制值;

压缩模块,用于将所述序号数组和所述差值数组压缩为差分包。

7.一种计算机装置,其特征在于,所述计算机装置包括处理器,所述处理器用于执行存储器中存储的计算机程序以实现如权利要求1-5中任一项所述差分包生成方法。

8.一种计算机存储介质,所述计算机存储介质上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1-5中任一项所述差分包生成方法。

差分包生成方法及相关设备

技术领域

[0001] 本发明涉及软件升级技术领域,具体涉及一种差分包生成方法、装置、计算机装置及计算机存储介质。

背景技术

[0002] 差分升级是电子设备软件升级的一种主要方式。用差分升级方法升级软件的主要过程为:计算出待升级软件的新文件和老文件的不同部分,将不同部分打包压缩成差分包(即升级补丁),将差分包发送给电子设备,电子设备将差分包和待升级软件的老文件合并成新文件。然而,现有的差分包生成算法生成的差分包较大,生成差分包耗费的时间较长。

发明内容

[0003] 鉴于以上内容,有必要提出一种差分包生成方法、装置、计算机装置及计算机存储介质,其可以生成用于软件升级的差分包。

[0004] 本申请的第一方面提供一种差分包生成方法,所述方法包括:

[0005] 获取待升级软件的老文件和新文件,所述老文件和所述新文件为文本文件;

[0006] 根据所述老文件中的分隔符将所述老文件划分为多个第一字符串,将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组;

[0007] 根据所述新文件中的分隔符将所述新文件划分为多个第二字符串,将所述多个第二字符串按照在所述新文件中的顺序组合为第二字符串组;

[0008] 从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串,将所述第二字符串组中的每个第二字符串的相似字符串在所述第一字符串组中的序号组合为序号数组;

[0009] 计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值,将所述第二字符串组中的每个第二字符串对应的差值组合为差值数组;

[0010] 将所述序号数组和所述差值数组压缩为差分包。

[0011] 另一种可能的实现方式中,所述将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组包括:

[0012] 删除所述多个第一字符串中重复的第一字符串,得到多个去重后的第一字符串;

[0013] 将所述多个去重后的第一字符串按照在所述老文件中的顺序组合为所述第一字符串组。

[0014] 另一种可能的实现方式中,所述从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串包括:

[0015] 对于所述第二字符串组中的任意一个第二字符串 A_i ,确定第二字符串 A_i 包含的字符的数量 m_i ,确定第二字符串 A_i 的各个字符 $a_{ix}, x=0, 1, 2, \dots, m_i-1$;

[0016] 对于所述第一字符串组中的任意一个第一字符串 B_j ,确定第一字符串 B_j 包含的字符

符的数量 n_j ,确定第一字符串 b_j 的各个字符 $b_{jy}, y=0, 1, 2, \dots, n_j-1$;

[0017] 计算第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的各个中间编辑距离 $v_{ij}(x+1, y+1), x=0, 1, 2, \dots, m_i-1, y=0, 1, 2, \dots, n_j-1$:

$$[0018] \quad v_{ij}(x+1, y+1) = \min \begin{cases} v_{ij}(x+1, y) \\ v_{ij}(x, y+1) \\ v_{ij}(x, y) + \begin{cases} 1 & \text{if } a_{ix} \neq b_{jy} \\ 0 & \text{if } a_{ix} = b_{jy} \end{cases} \end{cases}$$

[0019] 其中, $v_{ij}(x, 0) = x, v_{ij}(0, y) = y$, 第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的编辑距离为 $v_{ij}(m_i, n_j)$;

[0020] 从所述第一字符串组中查找与第二字符串 A_i 的编辑距离最小的第一字符串, 将所述编辑距离最小的第一字符串作为第二字符串 A_i 的相似字符串。

[0021] 另一种可能的实现方式中, 所述计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值包括:

[0022] 对所述第二字符串组中的每个第二字符串的二进制值和该第二字符串在所述第一字符串组中的相似字符串的二进制值进行对齐处理, 得到所述第二字符串组中的每个第二字符串的中间二进制值和该第二字符串在所述第一字符串组中的相似字符串的中间二进制值。

[0023] 另一种可能的实现方式中, 在所述获取待升级软件的老文件和新文件之前, 所述方法还包括:

[0024] 获取所述老文件的描述信息;

[0025] 根据所述老文件的描述信息判断所述老文件是否合法。

[0026] 另一种可能的实现方式中, 所述方法还包括:

[0027] 响应于客户端的下载请求, 将所述差分包发送给所述客户端, 使所述客户端根据所述差分包和所述老文件生成所述新文件。

[0028] 另一种可能的实现方式中, 所述方法还包括:

[0029] 将所述差分包分发送给服务器。

[0030] 本申请的第二方面提供一种差分包生成装置, 所述装置包括:

[0031] 获取模块, 用于获取待升级软件的老文件和新文件, 所述老文件和所述新文件为文本文件;

[0032] 第一划分模块, 用于根据所述老文件中的分隔符将所述老文件划分为多个第一字符串, 将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组;

[0033] 第二划分模块, 用于根据所述新文件中的分隔符将所述新文件划分为多个第二字符串, 将所述多个第二字符串按照在所述新文件中的顺序组合为第二字符串组;

[0034] 查找模块, 用于从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串, 将所述第二字符串组中的每个第二字符串的相似字符串在所述第一字符串组中的序号组合为序号数组;

[0035] 计算模块, 用于计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值, 将所述第二字符串组中的每个第二字符串对应的

差值组合为差值数组；

[0036] 压缩模块,用于将所述序号数组和所述差值数组压缩为差分包。

[0037] 另一种可能的实现方式中,所述将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组包括:

[0038] 删除所述多个第一字符串中重复的第一字符串,得到多个去重后的第一字符串;

[0039] 将所述多个去重后的第一字符串按照在所述老文件中的顺序组合为所述第一字符串组。

[0040] 另一种可能的实现方式中,所述从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串包括:

[0041] 对于所述第二字符串组中的任意一个第二字符串 A_i ,确定第二字符串 A_i 包含的字符的数量 m_i ,确定第二字符串 A_i 的各个字符 $a_{ix}, x=0, 1, 2, \dots, m_i-1$;

[0042] 对于所述第一字符串组中的任意一个第一字符串 B_j ,确定第一字符串 B_j 包含的字符的数量 n_j ,确定第一字符串 B_j 的各个字符 $b_{jy}, y=0, 1, 2, \dots, n_j-1$;

[0043] 计算第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的各个中间编辑距离 $v_{ij}(x+1, y+1), x=0, 1, 2, \dots, m_i-1, y=0, 1, 2, \dots, n_j-1$:

$$[0044] \quad v_{ij}(x+1, y+1) = \min \begin{cases} v_{ij}(x+1, y) \\ v_{ij}(x, y+1) \\ v_{ij}(x, y) + \begin{cases} 1 & \text{if } a_{ix} \neq b_{jy} \\ 0 & \text{if } a_{ix} = b_{jy} \end{cases} \end{cases}$$

[0045] 其中, $v_{ij}(x, 0) = x, v_{ij}(0, y) = y$, 第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的编辑距离为 $v_{ij}(m_i, n_j)$;

[0046] 从所述第一字符串组中查找与第二字符串 A_i 的编辑距离最小的第一字符串,将所述编辑距离最小的第一字符串作为第二字符串 A_i 的相似字符串。

[0047] 另一种可能的实现方式中,所述计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值包括:

[0048] 对所述第二字符串组中的每个第二字符串的二进制值和该第二字符串在所述第一字符串组中的相似字符串的二进制值进行对齐处理,得到所述第二字符串组中的每个第二字符串的中间二进制值和该第二字符串在所述第一字符串组中的相似字符串的中间二进制值。

[0049] 本申请的第三方面提供一种计算机装置,所述计算机装置包括处理器,所述处理器用于执行存储器中存储的计算机程序时实现所述差分包生成方法。

[0050] 本申请的第四方面提供一种计算机存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现所述差分包生成方法。

[0051] 相较于包括第二字符串组的差分包,本发明的差分包包括所述序号数组和所述差值数组,所占存储空间小;也就减少了生成本发明的差分包的时间,所以本发明减小了差分包的大小和提升了生成差分包的效率。

附图说明

- [0052] 图1是本发明实施例提供的差分包生成方法的流程图。
- [0053] 图2是本发明实施例提供的差分包生成装置的结构图。
- [0054] 图3是本发明实施例提供的计算机装置的示意图。

具体实施方式

[0055] 为了能够更清楚地理解本发明的上述目的、特征和优点,下面结合附图和具体实施例对本发明进行详细描述。需要说明的是,在不冲突的情况下,本申请的实施例及实施例中的特征可以相互组合。

[0056] 在下面的描述中阐述了很多具体细节以便于充分理解本发明,所描述的实施例仅仅是本发明一部分实施例,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,都属于本发明保护的范围。

[0057] 除非另有定义,本文所使用的所有的技术和科学术语与属于本发明的技术领域的技术人员通常理解的含义相同。本文中在本发明的说明书中所使用的术语只是为了描述具体的实施例的目的,不是旨在于限制本发明。

[0058] 优选地,本发明的差分包生成方法应用在一个或者多个计算机装置中。所述计算机装置是一种能够按照事先设定或存储的指令,自动进行数值计算和/或信息处理的设备,其硬件包括但不限于微处理器、专用集成电路(Application Specific Integrated Circuit,ASIC)、可编程门阵列(Field-Programmable Gate Array,FPGA)、数字处理器(Digital Signal Processor,DSP)、嵌入式设备等。

[0059] 所述计算机装置可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。所述计算机装置可以与用户通过键盘、鼠标、遥控器、触摸板或声控设备等方式进行人机交互。

[0060] 实施例一

[0061] 图1是本发明实施例一提供的差分包生成方法的流程图。所述差分包生成方法应用于计算机装置。

[0062] 所述差分包生成方法生成用于软件升级的差分包。

[0063] 如图1所示,所述差分包生成方法包括:

[0064] 101,获取待升级软件的老文件和新文件,所述老文件和所述新文件为文本文件。

[0065] 在一具体实施例中,可以从手机终端获取所述待升级软件的版本信息;从本地版本文件库查找版本信息与所述手机终端的所述待升级软件的版本信息一致的文件,得到所述待升级软件的所述老文件;从本地版本文件库查找所述待升级软件的最新版本的文件,得到所述待升级软件的所述新文件。本地版本文件库中存储所述待升级软件的多个历史版本和最新版本的文件。

[0066] 例如,从一个给定手机终端获取一个给定待升级软件的版本信息为V1.0.1;从本地版本文件库查找版本信息为V1.0.1的文件,得到给定待升级软件的版本信息为V1.0.1的老文件;从本地版本文件库查找给定待升级软件的最新版本的文件,得到给定待升级软件的版本信息为V1.0.2的新文件。本地版本文件库中可以存储给定待升级软件的版本信息为V1.0.0、V1.0.1、V1.0.2的文件。其中版本信息为V1.0.0、V1.0.1的文件为给定待升级软件

的两个历史版本的文件,版本信息为V1.0.2的文件为给定待升级软件的最新版本的文件。

[0067] 在另一具体实施例中,可以从电脑终端获取所述待升级软件的版本信息;从本地版本文件库查找版本信息与所述手机终端的所述待升级软件的版本信息一致的文件,得到所述待升级软件的所述老文件;从本地版本文件库查找所述待升级软件的最新版本的文件,得到所述待升级软件的所述新文件。

[0068] 在另一实施例中,在所述获取待升级软件的老文件和新文件之前,所述方法还包括:

[0069] 获取所述老文件的描述信息;

[0070] 根据所述老文件的描述信息判断所述老文件是否合法。

[0071] 例如,从客户端获取老文件的描述信息,客户端采用给定哈希算法对老文件进行哈希计算可以得到老文件的描述信息;对本地版本文件库中的老文件进行哈希计算得到本地的老文件的描述信息;若客户端的老文件的描述信息与本地的老文件的描述信息一致,则老文件合法;若客户端的老文件的描述信息与本地的老文件的描述信息不一致,则老文件不合法。

[0072] 102,根据所述老文件中的分隔符将所述老文件划分为多个第一字符串,将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组。

[0073] 所述老文件中的分隔符可以包括:换行符、空格符和制表符等。

[0074] 例如,根据老文件中的空格符将老文件划分为多个第一字符串,其中,老文件的内容为“I am not robot”,得到的多个第一字符串为{“I”,“am”,“not”,“robot”}。将多个第一字符串按照在所述老文件中的顺序组合为第一字符串组{b[1],b[2],b[3],b[4]},其中,b[1]=“I”,b[2]=“am”,b[3]=“not”,b[4]=“robot”。

[0075] 在一具体实施例中,所述将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组包括:

[0076] 删除所述多个第一字符串中重复的第一字符串,得到多个去重后的第一字符串;

[0077] 将所述多个去重后的第一字符串按照在所述老文件中的顺序组合为所述第一字符串组。

[0078] 例如,老文件的内容为“I am not robot,he is a robot”,得到的多个第一字符串为{“I”,“am”,“not”,“robot”,“,“,“he”,“is”,“a”,“robot”}。删除多个第一字符串中重复的第一字符串“robot”,得到多个去重后的第一字符串{“I”,“am”,“not”,“robot”,“,“,“he”,“is”,“a”},将多个去重后的第一字符串按照在所述老文件中的顺序组合为第一字符串组{b[1],b[2],b[3],b[4],b[5],b[6],b[7],b[8]},其中,b[1]=“I”,b[2]=“am”,b[3]=“not”,b[4]=“robot”,b[5]=“,“,b[6]=“he”,b[7]=“is”,b[8]=“a”。

[0079] 可以按照前后顺序保留所述重复的第一字符串中最前的第一字符串,删除所述重复的第一字符串中除最前的第一字符串之外的第一字符串。

[0080] 103,根据所述新文件中的分隔符将所述新文件划分为多个第二字符串,将所述多个第二字符串按照在所述新文件中的顺序组合为第二字符串组。

[0081] 所述新文件中的分隔符可以包括:换行符、空格符和制表符等。

[0082] 例如,根据新文件中的空格符将老文件划分为多个第一字符串,其中,新文件的内容为“I am a robot”,得到的多个第一字符串为{“I”,“am”,“a”,“robot”}。将多个第一字

字符串按照在所述老文件中的顺序组合为第一字符串组 $\{b[1], b[2], b[3], b[4]\}$, 其中, $b[1] = "I", b[2] = "am", b[3] = "a", b[4] = "robot"$ 。

[0083] 104, 从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串, 将所述第二字符串组中的每个第二字符串的相似字符串在所述第一字符串组中的序号组合为序号数组。

[0084] 所述从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串包括:

[0085] (1) 对于所述第二字符串组中的任意一个第二字符串 A_i , 确定第二字符串 A_i 包含的字符的数量 m_i , 确定第二字符串 A_i 的各个字符 $a_{ix}, x=0, 1, 2, \dots, m_i-1$ 。

[0086] (2) 对于所述第一字符串组中的任意一个第一字符串 B_j , 确定第一字符串 B_j 包含的字符的数量 n_j , 确定第一字符串 B_j 的各个字符 $b_{jy}, y=0, 1, 2, \dots, n_j-1$ 。

[0087] (3) 计算第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的各个中间编辑距离 $v_{ij}(x+1, y+1), x=0, 1, 2, \dots, m_i-1, y=0, 1, 2, \dots, n_j-1$:

$$[0088] \quad v_{ij}(x+1, y+1) = \min \begin{cases} v_{ij}(x+1, y) \\ v_{ij}(x, y+1) \\ v_{ij}(x, y) + \begin{cases} 1 & \text{if } a_{ix} \neq b_{jy} \\ 0 & \text{if } a_{ix} = b_{jy} \end{cases} \end{cases}$$

[0089] 其中, $v_{ij}(x, 0) = x, v_{ij}(0, y) = y$, 第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的编辑距离为 $v_{ij}(m_i, n_j)$ 。

[0090] 例如, 第二字符串组中的一个第二字符串为 "pingan", 第二字符串包含的字符的数量为 6, 第 0 个字符为 "p", 第 1 个字符为 "i", 第 2 个字符为 "n", 第 3 个字符为 "g", 第 4 个字符为 "a", 第 5 个字符为 "n"。第一字符串组中的一个第一字符串为 "pintai", 第一字符串包含的字符的数量为 6, 第 0 个字符为 "p", 第 1 个字符为 "i", 第 2 个字符为 "n", 第 3 个字符为 "t", 第 4 个字符为 "a", 第 5 个字符为 "i"。根据上述公式逐个计算中间编辑距离 $v(1, 2), v(2, 1), v(2, 2), v(2, 3), v(3, 2), v(3, 3), v(3, 4), v(4, 3), v(4, 4), v(4, 5), v(5, 4), v(5, 5), v(5, 6)$ 和 $v(6, 5)$ 。根据上述公式和 $v(5, 6), v(6, 5)$ 计算第二字符串与第一字符串的编辑距离 $v(6, 6)$ 。

[0091] (4) 从所述第一字符串组中查找与第二字符串 A_i 的编辑距离最小的第一字符串, 将所述编辑距离最小的第一字符串作为第二字符串 A_i 的相似字符串。

[0092] 105, 计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值, 将所述第二字符串组中的每个第二字符串对应的差值组合为差值数组。

[0093] 所述计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值包括:

[0094] 对所述第二字符串组中的每个第二字符串的二进制值和该第二字符串在所述第一字符串组中的相似字符串的二进制值进行对齐处理, 得到所述第二字符串组中的每个第二字符串的中间二进制值和该第二字符串在所述第一字符串组中的相似字符串的中间二进制值。例如, 若第二字符串的二进制值和该第二字符串的相似字符串的二进制值的长度不一致, 则长度长的二进制值不变, 在长度短的二进制值前逐个补零, 直到两个二进制值的

长度一致,得到长度一致的第二字符串的中间二进制值和第二字符串在第一字符串组中的相似字符串的中间二进制值。

[0095] 采用异或算法计算所述第二字符串组中的每个第二字符串的中间二进制值与该第二字符串的相似字符串的中间二进制值的差值。

[0096] 例如,一个第二字符串为“pingan”,该第二字符串的中间二进制值为“0000000111000011010011101110110011111000011101110”。该第二字符串的相似字符串为“pintai”,该第二字符串的相似字符串的中间二进制值为“1110000110100111011101100111111010011000011101001”。采用异或算法计算该第二字符串的中间二进制值与该第二字符串的相似字符串的中间二进制值的差值为“1110000001100100001110001001001001100000000000111”。

[0097] 106,将所述序号数组和所述差值数组压缩为差分包。

[0098] 可以采用bzip2算法将所述差值数组压缩为差分文件,将所述序号数组压缩为序号文件,将所述差分文件和序号文件合并为一个差分包。在压缩所述差值数组之前,可以将所述差值数组中的二进制值转换为十六进制值。

[0099] 在另一实施例中,所述序号数组可以不压缩,将所述差分文件和序号数组合并为一个差分包。

[0100] 当客户端接收到差分包时,可以对所述差分包进行解压,解压得到所述差值数组和所述序号数组。根据客户端中的所述老文件中的分隔符将所述老文件划分为多个字符串,将多个字符串按照在所述老文件中的顺序组合为第三字符串组,将所述第三字符串组的二进制值组合为二进制数组。根据所述序号数组查找与所述差值数组中每个二进制值对应的所述二进制数组中的二进制值,将所述差值数组中每个二进制值与该二进制值对应的所述二进制数组中的二进制值进行异或运算,可以得到所述新文件的数组,将所述新文件的数组转换为字符串,得到所述新文件。

[0101] 实施例一减小了所述差分包的大小和提升生成所述差分包的效率。

[0102] 在另一实施例中,所述方法还包括:

[0103] 响应于客户端的下载请求,将所述差分包发送给所述客户端,使所述客户端根据所述差分包和所述老文件生成所述新文件。

[0104] 所述差分包包括差分文件和序号文件,对所述差分包进行解压,可以得到所述差值数组和所述序号数组。客户都可以根据客户端中的所述老文件中的分隔符将所述老文件划分为多个字符串,将多个字符串按照在所述老文件中的顺序组合为第三字符串组,将所述第三字符串组的二进制值组合为二进制数组。根据所述序号数组查找与所述差值数组中每个二进制值对应的所述二进制数组中的二进制值,将所述差值数组中每个二进制值与该二进制值对应的所述二进制数组中的二进制值进行异或运算,可以得到所述新文件的数组,将所述新文件的数组转换为字符串,得到所述新文件。

[0105] 在另一实施例中,所述方法还包括:

[0106] 将所述差分包分发给服务器。

[0107] 在一具体实施例中,可以将所述差分包发送至各个分布式服务器中,当客户端请求所述差分包时,可以通过分布式服务器将所述差分包返回给客户端,增加客户端的下载效率。

[0108] 实施例二

[0109] 图2是本发明实施例二提供的差分包生成装置的结构图。所述差分包生成装置20应用于计算机装置。本装置的差分包生成是生成用于软件升级的差分包。如图2所示,所述差分包生成装置20可以包括获取模块201、第一划分模块202、第二划分模块203、查找模块204、计算模块205、压缩模块206。

[0110] 获取模块201,用于获取待升级软件的老文件和新文件,所述老文件和所述新文件为文本文件。

[0111] 在一具体实施例中,可以从手机终端获取所述待升级软件的版本信息;从本地版本文件库查找版本信息与所述手机终端的所述待升级软件的版本信息一致的文件,得到所述待升级软件的所述老文件;从本地版本文件库查找所述待升级软件的最新版本的文件,得到所述待升级软件的所述新文件。本地版本文件库中存储所述待升级软件的多个历史版本和最新版本的文件。

[0112] 例如,从一个给定手机终端获取一个给定待升级软件的版本信息为V1.0.1;从本地版本文件库查找版本信息为V1.0.1的文件,得到给定待升级软件的版本信息为V1.0.1的老文件;从本地版本文件库查找给定待升级软件的最新版本的文件,得到给定待升级软件的版本信息为V1.0.2的新文件。本地版本文件库中可以存储给定待升级软件的版本信息为V1.0.0、V1.0.1、V1.0.2的文件。其中版本信息为V1.0.0、V1.0.1的文件为给定待升级软件的两个历史版本的文件,版本信息为V1.0.2的文件为给定待升级软件的最新版本的文件。

[0113] 在另一具体实施例中,可以从电脑终端获取所述待升级软件的版本信息;从本地版本文件库查找版本信息与所述手机终端的所述待升级软件的版本信息一致的文件,得到所述待升级软件的所述老文件;从本地版本文件库查找所述待升级软件的最新版本的文件,得到所述待升级软件的所述新文件。

[0114] 在另一实施例中,在所述获取待升级软件的老文件和新文件之前,所述方法还包括:

[0115] 获取所述老文件的描述信息;

[0116] 根据所述老文件的描述信息判断所述老文件是否合法。

[0117] 例如,从客户端获取老文件的描述信息,客户端采用给定哈希算法对老文件进行哈希计算可以得到老文件的描述信息;对本地版本文件库中的老文件进行哈希计算得到本地的老文件的描述信息;若客户端的老文件的描述信息与本地的老文件的描述信息一致,则老文件合法;若客户端的老文件的描述信息与本地的老文件的描述信息不一致,则老文件不合法。

[0118] 第一划分模块202,用于根据所述老文件中的分隔符将所述老文件划分为多个第一字符串,将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组。

[0119] 所述老文件中的分隔符可以包括:换行符、空格符和制表符等。

[0120] 例如,根据老文件中的空格符将老文件划分为多个第一字符串,其中,老文件的内容为“I am not robot”,得到的多个第一字符串为{“I”,“am”,“not”,“robot”}。将多个第一字符串按照在所述老文件中的顺序组合为第一字符串组{b[1],b[2],b[3],b[4]},其中,b[1]=“I”,b[2]=“am”,b[3]=“not”,b[4]=“robot”。

[0121] 在一具体实施例中,所述将所述多个第一字符串按照在所述老文件中的顺序组合

为第一字符串组包括:

[0122] 删除所述多个第一字符串中重复的第一字符串,得到多个去重后的第一字符串;

[0123] 将所述多个去重后的第一字符串按照在所述老文件中的顺序组合为所述第一字符串组。

[0124] 例如,老文件的内容为“I am not robot,he is a robot”,得到的多个第一字符串为{“I”,“am”,“not”,“robot”,“,“,“he”,“is”,“a”,“robot”}。删除多个第一字符串中重复的第一字符串“robot”,得到多个去重后的第一字符串{“I”,“am”,“not”,“robot”,“,“,“he”,“is”,“a”},将多个去重后的第一字符串按照在所述老文件中的顺序组合为第一字符串组{b[1],b[2],b[3],b[4],b[5],b[6],b[7],b[8]},其中,b[1]=“I”,b[2]=“am”,b[3]=“not”,b[4]=“robot”,b[5]=“,“,b[6]=“he”,b[7]=“is”,b[8]=“a”。

[0125] 可以按照前后顺序保留所述重复的第一字符串中最前的第一字符串,删除所述重复的第一字符串中除最前的第一字符串之外的第一字符串。

[0126] 第二划分模块203,用于根据所述新文件中的分隔符将所述新文件划分为多个第二字符串,将所述多个第二字符串按照在所述新文件中的顺序组合为第二字符串组。

[0127] 所述新文件中的分隔符可以包括:换行符、空格符和制表符等。

[0128] 例如,根据新文件中的空格符将老文件划分为多个第一字符串,其中,新文件的内容为“I am a robot”,得到的多个第一字符串为{“I”,“am”,“a”,“robot”}。将多个第一字符串按照在所述老文件中的顺序组合为第一字符串组{b[1],b[2],b[3],b[4]},其中,b[1]=“I”,b[2]=“am”,b[3]=“a”,b[4]=“robot”。

[0129] 查找模块204,用于从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串,将所述第二字符串组中的每个第二字符串的相似字符串在所述第一字符串组中的序号组合为序号数组。

[0130] 所述从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串包括:

[0131] (1) 对于所述第二字符串组中的任意一个第二字符串 A_i ,确定第二字符串 A_i 包含的字符的数量 m_i ,确定第二字符串 A_i 的各个字符 $a_{ix},x=0,1,2,\dots,m_i-1$ 。

[0132] (2) 对于所述第一字符串组中的任意一个第一字符串 B_j ,确定第一字符串 B_j 包含的字符的数量 n_j ,确定第一字符串 B_j 的各个字符 $b_{jy},y=0,1,2,\dots,n_j-1$ 。

[0133] (3) 计算第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的各个中间编辑距离 $v_{ij}(x+1,y+1),x=0,1,2,\dots,m_i-1,y=0,1,2,\dots,n_j-1$:

$$[0134] \quad v_{ij}(x+1,y+1) = \min \begin{cases} v_{ij}(x+1,y) \\ v_{ij}(x,y+1) \\ v_{ij}(x,y) + \begin{cases} 1 & \text{if } a_{ix} \neq b_{jy} \\ 0 & \text{if } a_{ix} = b_{jy} \end{cases} \end{cases}$$

[0135] 其中, $v_{ij}(x,0)=x,v_{ij}(0,t)=t$,第二字符串 A_i 与所述第一字符串组中的每个第一字符串 B_j 的编辑距离为 $v_{ij}(m_i,n_j)$ 。

[0136] 例如,第二字符串组中的一个第二字符串为“pingan”,第二字符串包含的字符的数量为6,第0个字符为“p”,第1个字符为“i”,第2个字符为“n”,第3个字符为“g”,第4个字符

为“a”，第5个字符为“n”。第一字符串组中的一个第一字符串为“pintai”，第一字符串包含的字符的数量为6，第0个字符为“p”，第1个字符为“i”，第2个字符为“n”，第3个字符为“t”，第4个字符为“a”，第5个字符为“i”。根据上述公式逐个计算中间编辑距离 $v(1,2)$ 、 $v(2,1)$ 、 $v(2,2)$ 、 $v(2,3)$ 、 $v(3,2)$ 、 $v(3,3)$ 、 $v(3,4)$ 、 $v(4,3)$ 、 $v(4,4)$ 、 $v(4,5)$ 、 $v(5,4)$ 、 $v(5,5)$ 、 $v(5,6)$ 和 $v(6,5)$ 。根据上述公式和 $v(5,6)$ 、 $v(6,5)$ 计算第二字符串与第一字符串的编辑距离 $v(6,6)$ 。

[0137] (4) 从所述第一字符串组中查找与第二字符串 A_i 的编辑距离最小的第一字符串，将所述编辑距离最小的第一字符串作为第二字符串 A_i 的相似字符串。

[0138] 计算模块205，用于计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值，将所述第二字符串组中的每个第二字符串对应的差值组合为差值数组。

[0139] 所述计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值包括：

[0140] 对所述第二字符串组中的每个第二字符串的二进制值和该第二字符串在所述第一字符串组中的相似字符串的二进制值进行对齐处理，得到所述第二字符串组中的每个第二字符串的中间二进制值和该第二字符串在所述第一字符串组中的相似字符串的中间二进制值。例如，若第二字符串的二进制值和该第二字符串的相似字符串的二进制值的长度不一致，则长度长的二进制值不变，在长度短的二进制值前逐个补零，直到两个二进制值的长度一致，得到长度一致的第二字符串的中间二进制值和第二字符串在第一字符串组中的相似字符串的中间二进制值。

[0141] 采用异或算法计算所述第二字符串组中的每个第二字符串的中间二进制值与该第二字符串的相似字符串的中间二进制值的差值。

[0142] 例如，一个第二字符串为“pingan”，该第二字符串的中间二进制值为“0000000111000011010011101110110011111000011101110”。该第二字符串的相似字符串为“pintai”，该第二字符串的相似字符串的中间二进制值为“1110000110100111011101100111111010011000011101001”。采用异或算法计算该第二字符串的中间二进制值与该第二字符串的相似字符串的中间二进制值的差值为“111000000110010000111000100100100110000000000111”。

[0143] 压缩模块206，用于将所述序号数组和所述差值数组压缩为差分包。

[0144] 可以采用bz2算法将所述差值数组压缩为差分文件，将所述序号数组压缩为序号文件，将所述差分文件和序号文件合并为一个差分包。在压缩所述差值数组之前，可以将所述差值数组中的二进制值转换为十六进制值。

[0145] 在另一实施例中，所述序号数组可以不压缩，将所述差分文件和序号数组合并为一个差分包。

[0146] 当客户端接收到差分包时，可以对所述差分包进行解压，解压得到所述差值数组和所述序号数组。根据客户端中的所述老文件中的分隔符将所述老文件划分为多个字符串，将多个字符串按照在所述老文件中的顺序组合为第三字符串组，将所述第三字符串组的二进制值组合为二进制数组。根据所述序号数组查找与所述差值数组中每个二进制值对应的所述二进制数组中的二进制值，将所述差值数组中每个二进制值与该二进制值对应的所述二进制数组中的二进制值进行异或运算，可以得到所述新文件的数组，将所述新文件

的数组转换为字符串,得到所述新文件。

[0147] 实施例二的差分包生成装置20生成用于软件升级的差分包,减小了所述差分包的大小和提升生成所述差分包的效率。

[0148] 在另一实施例中,所述差分包生成装置20还可以包括:发送模块,用于响应于客户端的下载请求,将所述差分包发送给所述客户端,使所述客户端根据所述差分包和所述老文件生成所述新文件。

[0149] 所述差分包包括差分文件和序号文件,对所述差分包进行解压,可以得到所述差值数组和所述序号数组。客户都可以根据客户端中的所述老文件中的分隔符将所述老文件划分为多个字符串,将多个字符串按照在所述老文件中的顺序组合为第三字符串组,将所述第三字符串组的二进制值组合为二进制数组。根据所述序号数组查找与所述差值数组中每个二进制值对应的所述二进制数组中的二进制值,将所述差值数组中每个二进制值与该二进制值对应的所述二进制数组中的二进制值进行异或运算,可以得到所述新文件的数组,将所述新文件的数组转换为字符串,得到所述新文件。

[0150] 在另一实施例中,所述差分包生成装置20还可以包括:分发模块,用于将所述差分包分发给服务器。

[0151] 在一具体实施例中,可以将所述差分包发送至各个分布式服务器中,当客户端请求所述差分包时,可以通过分布式服务器将所述差分包返回给客户端,增加客户端的下载效率。

[0152] 实施例三

[0153] 本实施例提供一种计算机存储介质,该计算机存储介质上存储有计算机程序,该计算机程序被处理器执行时实现上述差分包生成方法实施例中的步骤,例如图1所示的101-106:

[0154] 101,获取待升级软件的老文件和新文件,所述老文件和所述新文件为文本文件;

[0155] 102,根据所述老文件中的分隔符将所述老文件划分为多个第一字符串,将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组;

[0156] 103,根据所述新文件中的分隔符将所述新文件划分为多个第二字符串,将所述多个第二字符串按照在所述新文件中的顺序组合为第二字符串组;

[0157] 104,从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串,将所述第二字符串组中的每个第二字符串的相似字符串在所述第一字符串组中的序号组合为序号数组;

[0158] 105,计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值,将所述第二字符串组中的每个第二字符串对应的差值组合为差值数组;

[0159] 106,将所述序号数组和所述差值数组压缩为差分包。

[0160] 或者,该计算机程序被处理器执行时实现上述装置实施例中各模块的功能,例如图2中的模块201-206:

[0161] 获取模块201,用于获取待升级软件的老文件和新文件,所述老文件和所述新文件为文本文件;

[0162] 第一划分模块202,用于根据所述老文件中的分隔符将所述老文件划分为多个第

一字符串,将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组;

[0163] 第二划分模块203,用于根据所述新文件中的分隔符将所述新文件划分为多个第二字符串,将所述多个第二字符串按照在所述新文件中的顺序组合为第二字符串组;

[0164] 查找模块204,用于从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串,将所述第二字符串组中的每个第二字符串的相似字符串在所述第一字符串组中的序号组合为序号数组;

[0165] 计算模块205,用于计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值,将所述第二字符串组中的每个第二字符串对应的差值组合为差值数组;

[0166] 压缩模块206,用于将所述序号数组和所述差值数组压缩为差分包。

[0167] 实施例四

[0168] 图3为本发明实施例四提供的计算机装置的示意图。所述计算机装置30包括存储器301、处理器302以及存储在所述存储器301中并可在所述处理器302上运行的计算机程序303,例如差分包生成程序。所述处理器302执行所述计算机程序303时实现上述差分包生成方法实施例中的步骤,例如图1所示的101-106:

[0169] 101,获取待升级软件的老文件和新文件,所述老文件和所述新文件为文本文件;

[0170] 102,根据所述老文件中的分隔符将所述老文件划分为多个第一字符串,将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组;

[0171] 103,根据所述新文件中的分隔符将所述新文件划分为多个第二字符串,将所述多个第二字符串按照在所述新文件中的顺序组合为第二字符串组;

[0172] 104,从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串,将所述第二字符串组中的每个第二字符串的相似字符串在所述第一字符串组中的序号组合为序号数组;

[0173] 105,计算所述第二字符串组中的每个第二字符串与该第二字符串在所述第一字符串组中的相似字符串的差值,将所述第二字符串组中的每个第二字符串对应的差值组合为差值数组;

[0174] 106,将所述序号数组和所述差值数组压缩为差分包。

[0175] 或者,该计算机程序被处理器执行时实现上述装置实施例中各模块的功能,例如图2中的模块201-206:

[0176] 获取模块201,用于获取待升级软件的老文件和新文件,所述老文件和所述新文件为文本文件;

[0177] 第一划分模块202,用于根据所述老文件中的分隔符将所述老文件划分为多个第一字符串,将所述多个第一字符串按照在所述老文件中的顺序组合为第一字符串组;

[0178] 第二划分模块203,用于根据所述新文件中的分隔符将所述新文件划分为多个第二字符串,将所述多个第二字符串按照在所述新文件中的顺序组合为第二字符串组;

[0179] 查找模块204,用于从所述第一字符串组中查找所述第二字符串组中的每个第二字符串的相似字符串,将所述第二字符串组中的每个第二字符串的相似字符串在所述第一字符串组中的序号组合为序号数组;

[0180] 计算模块205,用于计算所述第二字符串组中的每个第二字符串与该第二字符串

在所述第一字符串组中的相似字符串的差值,将所述第二字符串组中的每个第二字符串对应的差值组合为差值数组;

[0181] 压缩模块206,用于将所述序号数组和所述差值数组压缩为差分包。

[0182] 示例性的,所述计算机程序303可以被分割成一个或多个模块,所述一个或者多个模块被存储在所述存储器301中,并由所述处理器302执行,以完成本方法。所述一个或多个模块可以是能够完成特定功能的一系列计算机程序指令段,该指令段用于描述所述计算机程序303在所述计算机装置30中的执行过程。例如,所述计算机程序303可以被分割成图2中的获取模块201、第一划分模块202、第二划分模块203、查找模块204、计算模块205、压缩模块206,各模块具体功能参见实施例二。

[0183] 所述计算机装置30可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。本领域技术人员可以理解,所述示意图3仅仅是计算机装置30的示例,并不构成对计算机装置30的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件,例如所述计算机装置30还可以包括输入输出设备、网络接入设备、总线等。

[0184] 所称处理器302可以是中央处理单元(Central Processing Unit,CPU),还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器302也可以是任何常规的处理器等,所述处理器302是所述计算机装置30的控制中心,利用各种接口和线路连接整个计算机装置30的各个部分。

[0185] 所述存储器301可用于存储所述计算机程序303,所述处理器302通过运行或执行存储在所述存储器301内的计算机程序或模块,以及调用存储在存储器301内的数据,实现所述计算机装置30的各种功能。所述存储器301可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序(比如声音播放功能、图像播放功能等)等;存储数据区可存储根据计算机装置30的使用所创建的数据等。此外,存储器301可以包括非易失性存储器,例如硬盘、内存、插接式硬盘,智能存储卡(Smart Media Card,SMC),安全数字(Secure Digital,SD)卡,闪存卡(Flash Card)、至少一个磁盘存储器件、闪存器件、或其他非易失性固态存储器件。

[0186] 所述计算机装置30集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读存储介质中。基于这样的理解,本发明实现上述实施例方法中的全部或部分流程,也可以通过计算机程序来指令相关的硬件来完成,所述的计算机程序可存储于一计算机存储介质中,该计算机程序在被处理器执行时,可实现上述各个方法实施例的步骤。其中,所述计算机程序包括计算机程序代码,所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质可以包括:能够携带所述计算机程序代码的任何实体或装置、记录介质、U盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器(ROM,Read-Only Memory)。

[0187] 在本发明所提供的几个实施例中,应该理解到,所揭露的系统,装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述模块的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式。

[0188] 所述作为分离部件说明的模块可以是或者也可以不是物理上分开的,作为模块显示的部件可以是或者也可以不是物理模块,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。

[0189] 另外,在本发明各个实施例中的各功能模块可以集成在一个处理模块中,也可以是各个模块单独物理存在,也可以两个或两个以上模块集成在一个模块中。上述集成的模块既可以采用硬件的形式实现,也可以采用硬件加软件功能模块的形式实现。

[0190] 上述以软件功能模块的形式实现的集成的模块,可以存储在一个计算机可读取存储介质中。上述软件功能模块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(可以是个人计算机,服务器,或者网络设备等)或处理器(processor)执行本发明各个实施例所述方法的部分步骤。

[0191] 对于本领域技术人员而言,显然本发明不限于上述示范性实施例的细节,而且在不背离本发明的精神或基本特征的情况下,能够以其他的具体形式实现本发明。因此,无论从哪一点来看,均应将实施例看作是示范性的,而且是非限制性的,本发明的范围由所附权利要求而不是上述说明限定,因此旨在将落在权利要求的等同要件的含义和范围内的所有变化涵括在本发明内。不应将权利要求中的任何附图标记视为限制所涉及的权利要求。此外,显然“包括”一词不排除其他模块或步骤,单数不排除复数。系统权利要求中陈述的多个模块或装置也可以由一个模块或装置通过软件或者硬件来实现。第一,第二等词语用来表示名称,而并不表示任何特定的顺序。

[0192] 最后应说明的是,以上实施例仅用以说明本发明的技术方案而非限制,尽管参照较佳实施例对本发明进行了详细说明,本领域的普通技术人员应当理解,可以对本发明的技术方案进行修改或等同替换,而不脱离本发明技术方案的精神和范围。

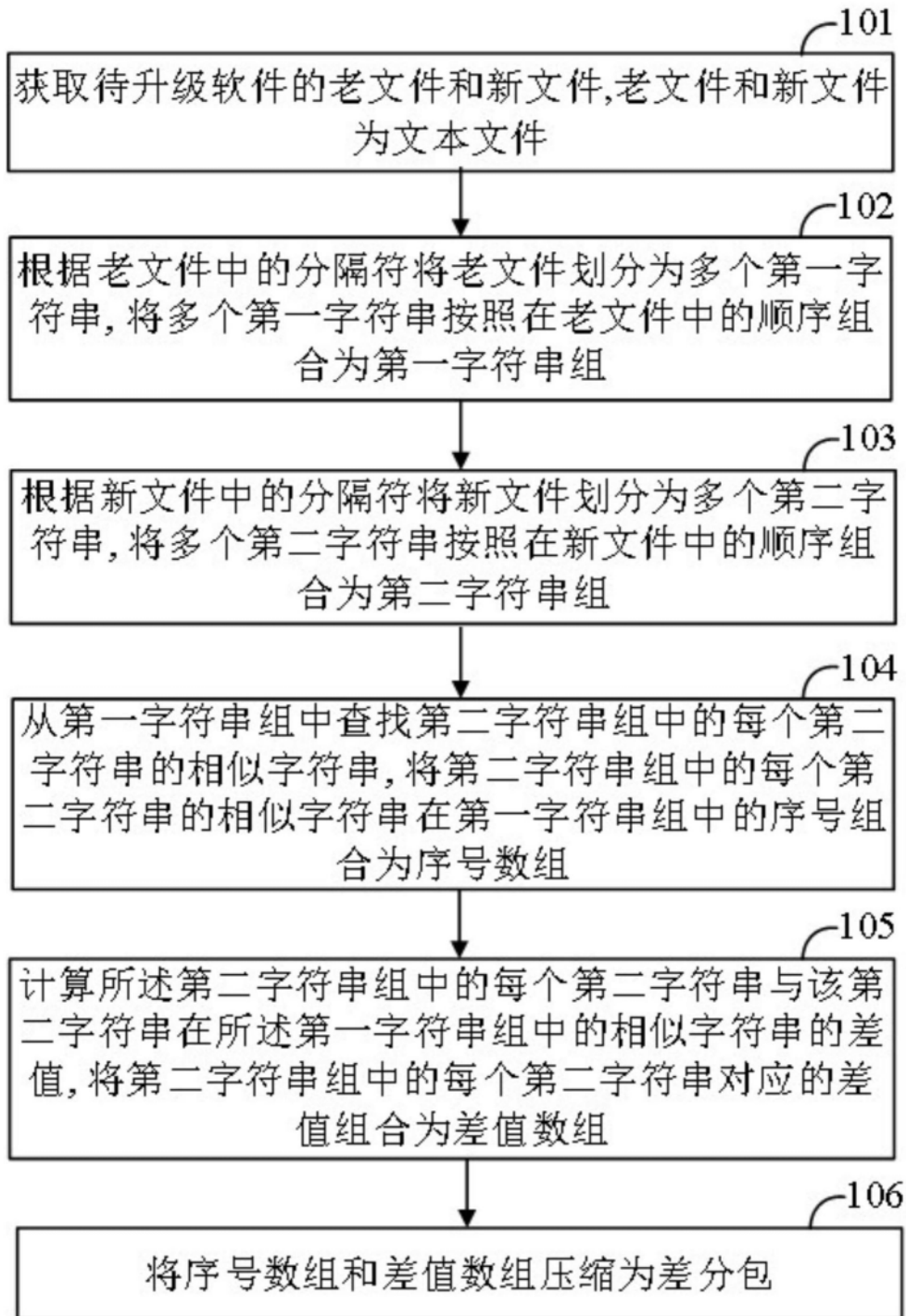


图1

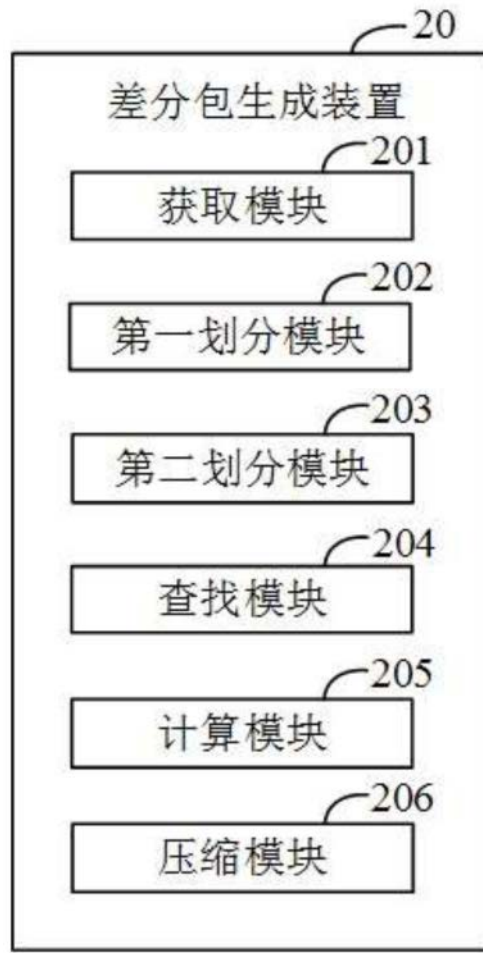


图2

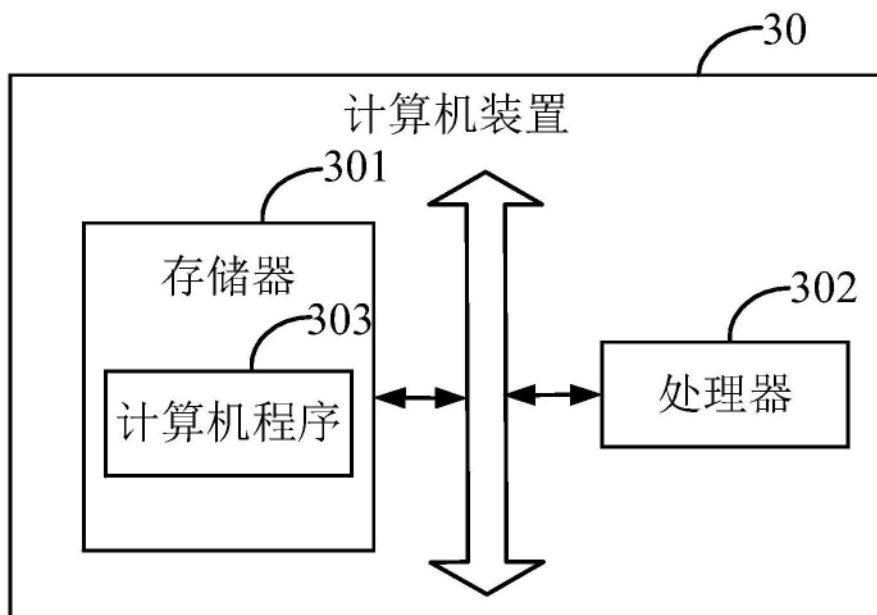


图3