



(12) 发明专利

(10) 授权公告号 CN 110866492 B

(45) 授权公告日 2022.12.13

(21) 申请号 201911107290.9

CN 101763248 A, 2010.06.30

(22) 申请日 2019.11.13

CN 106999076 A, 2017.08.01

(65) 同一申请的已公布的文献号

CN 104203340 A, 2014.12.10

申请公布号 CN 110866492 A

CN 108764065 A, 2018.11.06

(43) 申请公布日 2020.03.06

CN 110414237 A, 2019.11.05

(73) 专利权人 广州品唯软件有限公司

CN 108885660 A, 2018.11.23

地址 510145 广东省广州市荔湾区芳村大道中314号自编之一

CN 104317843 A, 2015.01.28

CN 107193739 A, 2017.09.22

CN 106055984 A, 2016.10.26

CN 109960643 A, 2019.07.02

(72) 发明人 陈策

CN 101932999 A, 2010.12.29

(74) 专利代理机构 北京市万慧达律师事务所

CN 108446135 A, 2018.08.24

11111

CN 102855131 A, 2013.01.02

专利代理师 顾友

CN 103853556 A, 2014.06.11

(51) Int. Cl.

CN 109725926 A, 2019.05.07

G06V 30/412 (2022.01)

CN 109783128 A, 2019.05.21

(56) 对比文件

CN 109783128 A, 2019.05.21

CN 103006205 A, 2013.04.03

CN 105193451 A, 2015.12.30

孙洁. 支持软件配置管理的软件开发过程研究及应用.《中国优秀硕士学位论文全文数据库信息科技辑》.2006,第5-79页.

审查员 路寒冰

权利要求书1页 说明书7页 附图3页

(54) 发明名称

一种基线分支的识别方法、装置及计算机系统

(57) 摘要

本申请实施例公开了一种基线识别的方法、装置及计算机系统,所述方法包括:调用预先配置的接口,获得待测分支的提交记录,所述提交记录包含每一所述待测分支的子分支记录;根据每一所述待测分支的子分支记录,计算每一所述待测分支的子分支个数;确定所述子分支个数在预设阈值以上的待测分支为基线分支,无需人工进行识别,实现了对基线分支的自动识别,提高了基线分支的识别效率,保证了识别的准确性。



1. 一种基线分支的识别方法,其特征在于,所述方法包括:

识别待测版本号,自动查找所述版本号的对应的项目;

调用预先配置的接口,获取所述对应的项目的需求列表,根据所述需求列表查找对应的待测分支,并获得所述待测分支的提交记录,所述提交记录包含每一所述待测分支的子分支记录;

根据每一所述待测分支的子分支记录,按照所述提交记录中所述待测分支的提交顺序,倒序计算每一所述待测分支的子分支个数;

确定所述子分支个数在预设阈值以上的所述待测分支为基线分支;所述预设阈值为2;

对所述基线分支进行标记并记录,并将所述记录数据持久化至数据库;

将所述记录进行可视化展示。

2. 根据权利要求1所述的方法,其特征在于,确定所述子分支个数在预设阈值以上的待测分支为基线分支后,所述方法还包括:

确定所述子分支个数小于所述预设阈值的所述待测分支为未测分支。

3. 一种基线分支识别装置,其特征在于,所述装置包括:

获取模块,识别待测版本号,自动查找所述版本号的对应的项目;

用于调用预先配置的接口,获取所述对应的项目的需求列表,根据所述需求列表查找对应的待测分支,并获得所述待测分支的提交记录,所述待测分支的提交记录包含每一所述待测分支的子分支记录;

计算模块,用于根据每一所述待测分支的子分支记录,按照所述提交记录中所述待测分支的提交顺序,倒序计算每一所述待测分支的子分支个数;

处理模块,用于确定所述子分支个数在预设阈值以上的所述待测分支为基线分支;所述预设阈值为2;

记录模块,用于对所述基线分支进行标记并记录,并将所述记录数据持久化至数据库,并将所述记录进行可视化展示。

4. 一种计算机系统,其特征在于,所述系统包括:

一个或多个处理器;

以及与所述一个或多个处理器关联的存储器,所述存储器用于存储程序指令,所述程序指令在被所述一个或多个处理器读取执行时,执行如权利要求1所述的方法的步骤。

一种基线分支的识别方法、装置及计算机系统

技术领域

[0001] 本发明涉及测试领域,尤其涉及一种基线分支的识别方法、装置及计算机系统。

背景技术

[0002] 基线是软件文档或源码的一个稳定的版本,只有测试评审通过的文档或产品才能被纳入基线,是进一步开发的基础。基线由基线分支组成,在每个单独的基线分支基础上,可以开发新的分支。为了确保新分支投入生产环境后不会产生问题,需要对这些新的分支进行测试。

[0003] 当需要对新的分支进行测试时,需要预先了解这些新分支对应的基线分支,才能在基线分支已实现的功能基础上,对包含了基线分支及新的分支的待测分支进行测试。在现有技术中,通常都是需要人工进行基线分支的识别,对测试人员的代码能力有着很高的要求,而且人工辨认的效率低,容易出错。

发明内容

[0004] 为了解决现有技术的不足,本发明的主要目的在于提供一种基线分支的识别方法、装置及计算机系统,以实现自动识别,提高识别效率。

[0005] 第一方面,本发明提供了一种基线分支的识别方法,所述方法包括:

[0006] 调用预先配置的接口,获得待测分支的提交记录,所述提交记录包含每一所述待测分支的子分支记录;

[0007] 根据每一所述待测分支的子分支记录,计算每一所述待测分支的子分支个数;

[0008] 确定所述子分支个数在预设阈值以上的所述待测分支为基线分支。

[0009] 在一些实施例中,所述预设阈值为2。

[0010] 在一些实施例中,所述根据每一所述待测分支的子分支记录,计算每一所述待测分支的子分支个数包括:

[0011] 根据每一所述待测分支的子分支记录,按照所述提交记录中所述待测分支的提交顺序,倒序计算每一所述待测分支的子分支个数。

[0012] 在一些实施例中,确定所述子分支个数在预设阈值以上的所述待测分支为基线分支后,所述方法还包括:

[0013] 对所述基线分支进行标记并记录,并将所述记录数据持久化至数据库。

[0014] 在一些实施例中,将所述记录数据持久化至数据库后,所述方法还包括将所述记录进行可视化展示。

[0015] 在一些实施例中,确定所述子分支个数在预设阈值以上的所述待测分支为基线分支后,所述方法还包括:

[0016] 确定所述子分支个数小于所述预设阈值的所述待测分支为未测分支。

[0017] 第二方面,本发明提供了一种基线分支识别装置,所述装置包括:

[0018] 获取模块,用于调用预先配置的接口,获得待测分支的提交记录,所述待测分支的

提交记录包含每一待测分支的子分支记录；

[0019] 计算模块,用于根据每一所述待测分支的子分支记录,计算每一所述待测分支的子分支个数；

[0020] 处理模块,用于确定所述子分支个数在预设阈值以上的所述待测分支为基线分支。

[0021] 在一些实施例中,所述计算模块还可用于根据每一所述待测分支的子分支记录,按照所述提交记录中所述待测分支的提交顺序,倒序计算每一所述待测分支的子分支个数。

[0022] 在一些实施例中,所述装置还包括记录模块,用于对所述基线分支进行标记并记录,并将所述记录数据持久化至数据库。

[0023] 第三方面,本申请提供了一种计算机系统,所述系统包括:

[0024] 一个或多个处理器；

[0025] 以及与所述一个或多个处理器关联的存储器,所述存储器用于存储程序指令,所述程序指令在被所述一个或多个处理器读取执行时,执行如下操作:

[0026] 调用预先配置的接口,获得待测分支的提交记录,所述提交记录包含每一所述待测分支的子分支记录；

[0027] 根据每一所述待测分支的子分支记录,计算每一所述待测分支的子分支个数；

[0028] 确定所述子分支个数在预设阈值以上的所述待测分支为基线分支。

[0029] 根据本申请提供的具体实施例,本申请公开了以下技术效果:

[0030] 本申请提出了通过调用预先配置的接口,获得所有待测分支的子分支记录,通过计算每一待测分支的子分支个数,将子分支个数在预设阈值以上的分支确定基线分支的方法,实现了对基线分支的自动识别,提高了基线分支的识别效率,并保证了识别的准确性；

[0031] 本申请提出了根据每一所述待测分支的子分支记录,按照所述提交记录中所述待测分支的提交顺序,倒序计算每一所述待测分支的子分支个数的方法,提高了子分支个数的计算效率；

[0032] 进一步地,本申请提出了将确定为基线分支的待测分支进行标记并记录,并数据持久化至数据库的方法,方便了技术人员基于识别的基线分支进行下一步的测试操作。

[0033] 当然,实施本申请的任一产品并不一定需要同时达到以上所述的所有优点。

附图说明

[0034] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例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发明的一些实施例,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,还可以根据这些附图获得其他的附图。

[0035] 图1是本申请实施例提供的项目分支结构图；

[0036] 图2是本申请实施例提供的方法流程图；

[0037] 图3是本申请实施例提供的装置结构图；

[0038] 图4是本申请实施例提供的计算机结构图。

具体实施方式

[0039] 为使本发明的目的、技术方案和优点更加清楚,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。

[0040] 如背景技术中所述,基线分支是进一步开发的基础。为了实现新的需求,会在基线的基础下衍生出新开发的分支,而这些新开发的分支需要经过测试评审,才能投入实际应用,并进一步地被加入至基线中。

[0041] 图1示出了项目的分支结构图。在一次分支测试中,一个完整的待测分支包含了多个子待测分支,这些子待测分支可分为基线分支和未测分支。基线分支是基线的一部分,是已经评审通过了的稳定版本,未测分支则代表着新开发的未通过测试评审的新的代码模块。对待测分支进行的分支测试是在基线分支的功能基础上进行的。

[0042] 因此,为了对这些新分支进行测试,需要预先识别待测分支中的基线分支,在基线分支的功能基础上,对新分支进行测试。然而,当开发人员未对基线分支预先标明的情况下,测试人员如果要对待测分支进行测试,就必须对基线分支进行识别。

[0043] 在现有技术中,通常通过测试人员人工识别的方式来确定待测分支的基线分支,对测试人员的代码水平要求高、耗费时间长且容易出错。

[0044] 为解决上述问题,本申请提供了一种基线分支的识别方法,通过调用预先配置的接口,获得所有待测分支的提交记录,所述提交记录中包含了每一待测分支的子节点的记录;然后根据提交记录中每一待测分支的提交顺序,倒序计算每一待测分支的子节点个数。当计算得到一个分支的子节点个数不少于2个时,确定该分支为基线分支,并进一步地将其标记为基线分支并进行记录,然后将该记录数据持久化至数据库中,供测试人员参考,实现了基线分支的自动识别,提高了基线的识别效率。

[0045] 实施例一

[0046] 具体的,上述方法可通过如下步骤实现:

[0047] 步骤一、识别待测版本号,使用Dashboard自动查找所述版本号的对应的项目;

[0048] Dashboard是一种支持数据可视化的数据管理工具,可从多个数据源实时获取数据。使用Dashboard可对每一版本号包含的所有的项目进行记录,供需要时调用。

[0049] 通过识别待测试的版本号,可使用Dashboard调取该版本号对应的项目,根据该项目调用预先配置的接口获取对应的需求列表。

[0050] 步骤二、调用预先配置的接口,获取所述对应的项目的需求列表,根据所述需求列表查找对应的待测分支,并获得对应待测分支的提交记录,所述提交记录包括每一待测分支包含的commitid记录;

[0051] 该接口可以是预先配置的项目管理接口,提供了获取需求列表、待测分支的提交记录、项目与开发人员的对应关系、项目与测试人员的对应关系等功能。

[0052] 如图1所示,在进行分支测试时,不需要对整个完整的项目进行完整的测试,可查找待测项目的需求列表,获得待测需求,根据待测需求自动查找对应的待测分支,对包含了一个基线分支及一个或多个未测分支的待测分支进行测试。

[0053] 在一个分支被提交时,该次提交会被自动分配一个commitid。commitid是一个随

机产生的字符串,每次提交产生的commitid都不相同,根据commitid可查找到对应的被提交的分支。

[0054] 在原始的分支的基础上开发出的新分支,也会包含该原始的分支的commitid。该原始的分支可以称为父分支,新分支可以称为子分支。子分支在需要时可以通过查找其包含的commitid被分配的分支来找到它的父分支。

[0055] 步骤三、根据每一待测分支的包含的commitid,按照提交记录中待测分支的提交顺序,倒序计算每一待测分支的子分支的个数;

[0056] 由于每个子分支都会包含父分支的commitid,因此可以统计包含某个commitid的分支的个数,来计算被分配该commitid的分支的子分支个数。

[0057] 例如,当一个分支被分配的commitid为1001时,若在该分支的基础上,又开发了3个新的分支,则当计算该分支的子分支个数时,可以查找包含了1001的分支,并对包含1001的分支的个数进行统计,统计获得的个数即为该分支的子分支个数。由于该分支本身同样包含了1001的记录,因此该分支本身也会被纳入统计,最终得到的子分支个数为4。

[0058] 而对没有子分支的分支而言,在计算该分支的子分支个数时,它被分配的commitid只能在一个分支内被查找到,因此它的子分支个数为1。

[0059] 优选的,可根据按照提交记录中每一待测分支的提交顺序,倒序计算每一待测分支的子分支个数。

[0060] 步骤四、确定子分支个数不少于2个的待测分支为基线分支,确定子节点个数少于2个的待测分支为未测分支;

[0061] 如上所述,待测分支包含了基线分支和未测分支。基线分支组成了基线,代表着稳定的功能基础。每个未测分支都代表着新开发的功能或需求,需要经过测试评审才能投入生产。在一个稳定的基线分支的基础上,才能开发新的分支即未测分支,所以每个未测分支都对应着一个单独的基线分支。

[0062] 对于尚不稳定的未测分支,它不能作为开发新的分支的基础,它的子分支数应当为1;对于基线分支而言,它本身已经是稳定的版本了,只是因为在其的基础上又开发了新分支,才又需要对其进行分支测试,所以当基线分支成为待测分支时,如上所述,它的子分支个数应当至少为2,包括了它本身以及在其基础上开发的未测分支。

[0063] 基于基线分支的这一特性,我们可以根据每个待测分支的子分支个数,确定每个待测分支是否为基线分支。

[0064] 为了将基线的识别结果进行保存,方便后续根据基线分支对待测分支进行测试,需进行:

[0065] 步骤五、对所述基线分支进行标记并记录,并将所述记录数据持久化至数据库;

[0066] 可将基线分支的名称、commitid等,生成数据库数据记录,并将该记录数据持久化至数据库,在分支测试时需要时可查找数据库获得该记录。

[0067] 步骤六、对待测分支进行测试。

[0068] 具体的,还可根据基线分支及已测试的测试用例覆盖的未测分支,计算获得测试覆盖率结果。

[0069] 优选的,还可通过调用接口,获取项目与测试人员的对应关系,并进一步根据项目与待测分支的对应关系,确定待测分支与测试人员的对应关系,在识别基线分支之后将待

测分支分配给对应的测试人员。

[0070] 优选的,还可通过调用接口,获取项目与开发人员的对应关系,并进一步根据项目与待测分支的对应关系,确定待测分支与开发人员的对应关系,在对待测分支进行测试后将测试结果发送至对应的开发人员,供开发人员参考。

[0071] 实施例二

[0072] 对应上述实施例一,本申请提供了一种基线的识别方法,如图2所示,所述方法包括:

[0073] 210、调用预先配置的接口,获得待测分支的提交记录,所述提交记录包含每一所述待测分支的子分支记录;

[0074] 优选的,可调用预先配置的接口,获取所述待测项目的需求列表,根据所述需求列表查找获得对应的待测分支的提交记录。

[0075] 220、根据每一所述待测分支的子分支记录,计算每一所述待测分支的子分支个数;

[0076] 优选的,所述子分支为包含所述待测分支的commitid的分支。

[0077] 优选的,可根据包含每一待测分支的commitid的分支个数,计算每一待测分支的子分支个数。

[0078] 优选的,所述根据每一所述待测分支的子分支记录,计算每一所述待测分支的子分支个数还包括:

[0079] 221、根据每一所述待测分支的子分支记录,按照所述提交记录中所述待测分支的提交顺序,倒序计算每一所述待测分支的子分支个数。

[0080] 230、确定所述子分支个数在预设阈值以上的所述待测分支为基线分支;

[0081] 优选的,所述预设阈值为2。

[0082] 240、确定所述子分支个数小于所述预设阈值的待测分支为未测分支;

[0083] 250、对所述基线分支进行标记并记录,并将所述记录数据持久化至数据库;

[0084] 260、将所述记录进行可视化展示。

[0085] 优选的,可根据所述记录生成基线分支报告,供技术人员参考。

[0086] 优选的,还可通过调用接口,获取待测分支与测试人员的对应关系,在确定基线分支之后将待测分支分配给对应的测试人员。

[0087] 优选的,还可通过调用接口,获取待测分支与开发人员的对应关系,在对待测分支进行测试后将测试结果发送至对应的开发人员,供开发人员参考。

[0088] 实施例三

[0089] 对应上述实施例二,本申请提供了一种基线识别装置,如图3所示,所述装置包括:

[0090] 获取模块310,用于调用预先配置的接口,获得待测分支的提交记录,所述待测分支的提交记录包含每一待测分支的子分支记录;

[0091] 计算模块320,用于根据每一所述待测分支的子分支记录,计算每一所述待测分支的子分支个数;

[0092] 处理模块330,用于确定所述子分支个数在预设阈值以上的分支为基线分支。

[0093] 优选的,所述计算模块还可用于根据每一所述待测分支的子分支记录,按照所述提交记录中所述待测分支的提交顺序,倒序计算每一所述待测分支的子分支个数。

[0094] 优选的,所述计算模块320还可用于根据每一所述待测分支的子分支记录,按照所述提交记录中所述待测分支的提交顺序,倒序计算每一所述待测分支的子分支个数。

[0095] 优选的,所述装置还包括记录模块340,用于对所述基线分支进行标记并记录,并将所述记录数据持久化至数据库。

[0096] 优选的,所述处理模块330还可用于确定所述子分支个数在2以上的分支为基线分支。

[0097] 优选的,所述获取模块还可用于用于调用预先配置的接口,获得待测分支与开发人员的对应关系;所述装置还包括分配模块350,用于在确定所述子分支个数在预设阈值以上的分支为基线分支后,将所述待测分支发送给对应的测试人员。

[0098] 实施例四

[0099] 对应上述实施例,本申请还提供了一种计算机系统,包括一个或多个处理器;以及与所述一个或多个处理器关联的存储器,所述存储器用于存储程序指令,所述程序指令在被所述一个或多个处理器读取执行时,执行如下操作:

[0100] 调用预先配置的项目管理接口,获得待测分支的提交记录,所述提交记录包含每一所述待测分支的子分支记录;

[0101] 根据每一所述待测分支的子分支记录,计算每一所述待测分支的子分支个数;

[0102] 确定所述子分支个数在预设阈值以上的所述待测分支为基线分支。

[0103] 其中,图4示例性的展示出了计算机系统的架构,具体可以包括处理器1510,视频显示适配器1511,磁盘驱动器1512,输入/输出接口1513,网络接口1514,以及存储器1520。上述处理器1510、视频显示适配器1511、磁盘驱动器1512、输入/输出接口1513、网络接口1514,与存储器1520之间可以通过通信总线1530进行通信连接。

[0104] 其中,处理器1510可以采用通用的CPU(Central Processing Unit,中央处理器)、微处理器、应用专用集成电路(Application Specific Integrated Circuit,ASIC)、或者一个或多个集成电路等方式实现,用于执行相关程序,以实现本申请所提供的技术方案。

[0105] 存储器1520可以采用ROM(Read Only Memory,只读存储器)、RAM(Random Access Memory,随机存取存储器)、静态存储设备,动态存储设备等形式实现。存储器1520可以存储用于控制计算机系统1500运行的操作系统1521,用于控制计算机系统1500的低级别操作的基本输入输出系统(BIOS)。另外,还可以存储网页浏览器1523,数据存储管理系统1524,以及图标字体处理系统1525等等。上述图标字体处理系统1525就可以是本申请实施例中具体实现前述各步骤操作的应用程序。总之,在通过软件或者固件来实现本申请所提供的技术方案时,相关的程序代码保存在存储器1520中,并由处理器1510来调用执行。

[0106] 输入/输出接口1513用于连接输入/输出模块,以实现信息输入及输出。输入输出/模块可以作为组件配置在设备中(图中未示出),也可以外接于设备以提供相应功能。其中输入设备可以包括键盘、鼠标、触摸屏、麦克风、各类传感器等,输出设备可以包括显示器、扬声器、振动器、指示灯等。

[0107] 网络接口1514用于连接通信模块(图中未示出),以实现本设备与其他设备的通信交互。其中通信模块可以通过有线方式(例如USB、网线等)实现通信,也可以通过无线方式(例如移动网络、WIFI、蓝牙等)实现通信。

[0108] 总线1530包括一通路,在设备的各个组件(例如处理器1510、视频显示适配器

1511、磁盘驱动器1512、输入/输出接口1513、网络接口1514,与存储器1520)之间传输信息。

[0109] 另外,该计算机系统1500还可以从虚拟资源对象领取条件信息数据库1541中获得具体领取条件的信息,以用于进行条件判断,等等。

[0110] 需要说明的是,尽管上述设备仅示出了处理器1510、视频显示适配器1511、磁盘驱动器1512、输入/输出接口1513、网络接口1514,存储器1520,总线1530等,但是在具体实施过程中,该设备还可以包括实现正常运行所必需的其他组件。此外,本领域的技术人员可以理解的是,上述设备中也可以仅包含实现本申请方案所必需的组件,而不必包含图中所示的全部组件。

[0111] 通过以上的实施方式的描述可知,本领域的技术人员可以清楚地了解到本申请可借助软件加必需的通用硬件平台的方式来实现。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品可以存储在存储介质中,如ROM/RAM、磁碟、光盘等,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,云服务器,或者网络设备等)执行本申请各个实施例或者实施例的某些部分所述的方法。

[0112] 以上所述仅为本发明的较佳实施例,并不用以限制本发明,凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

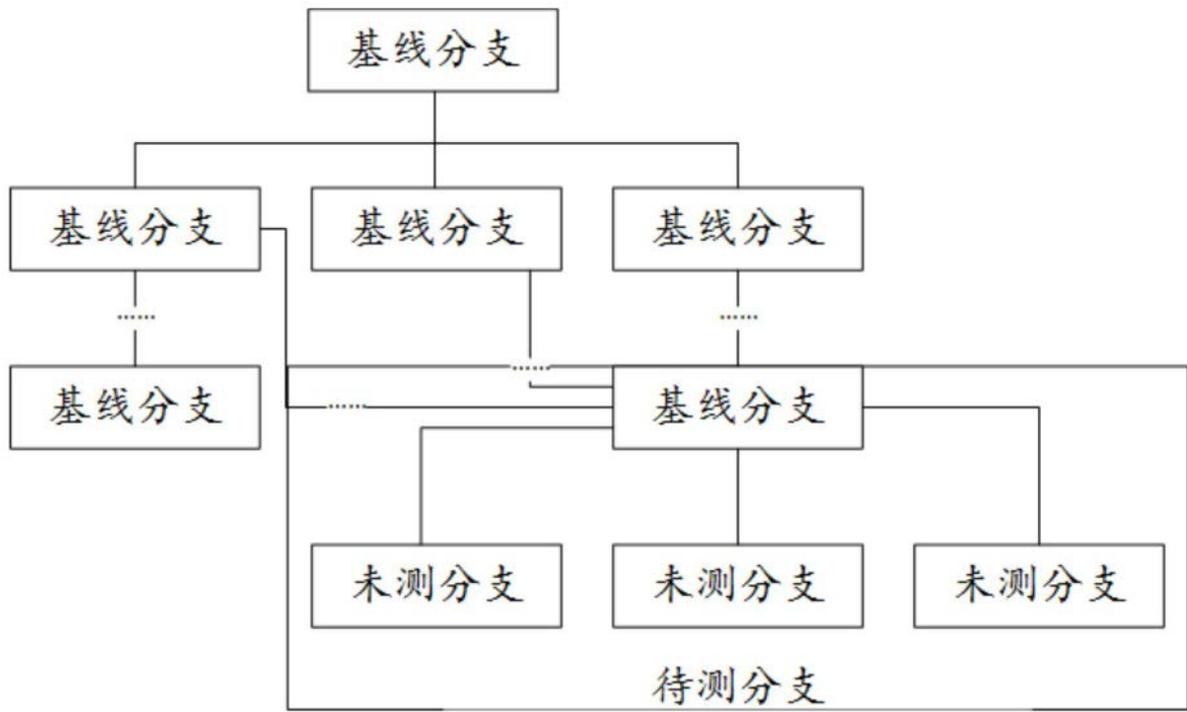


图1

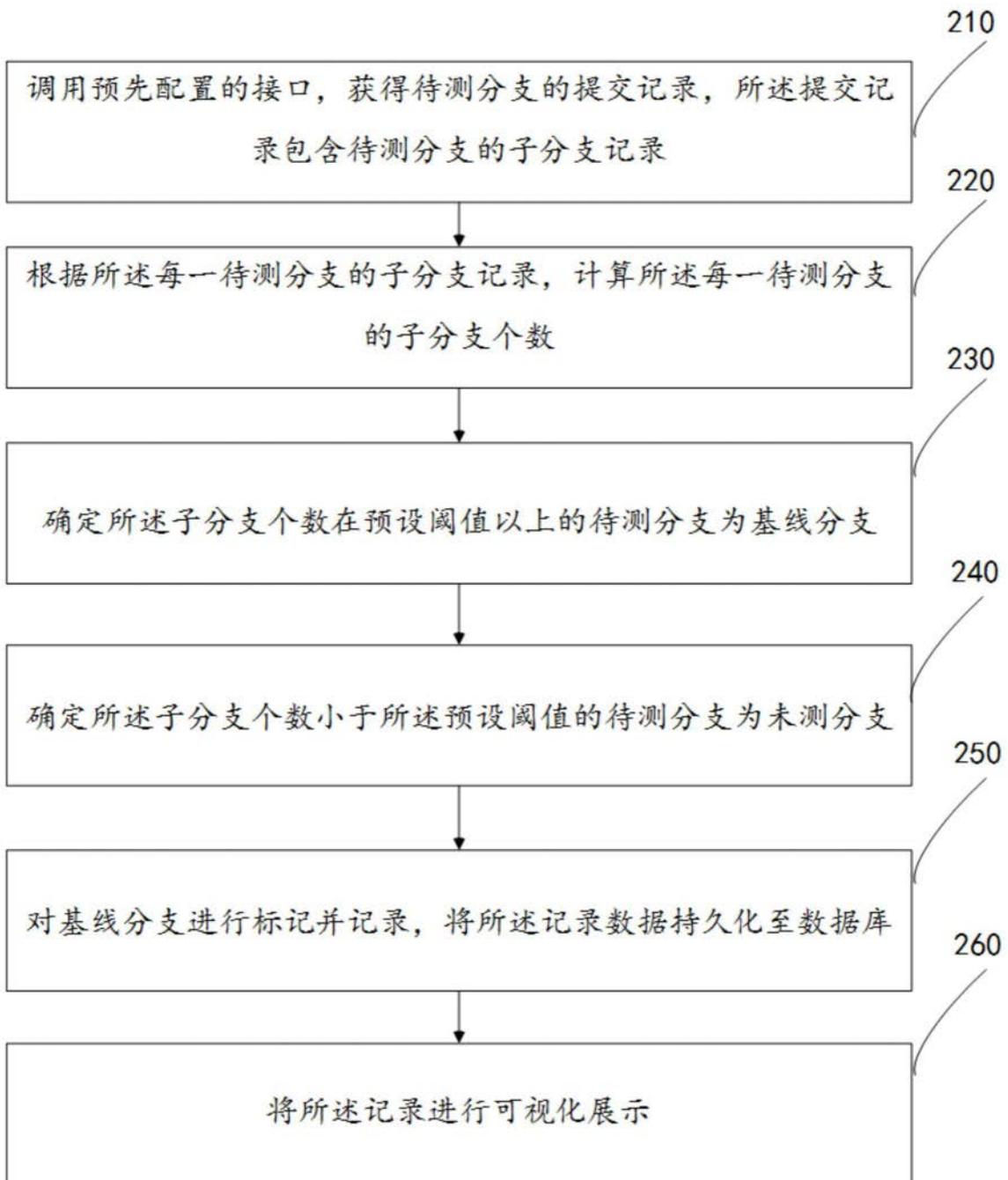


图2

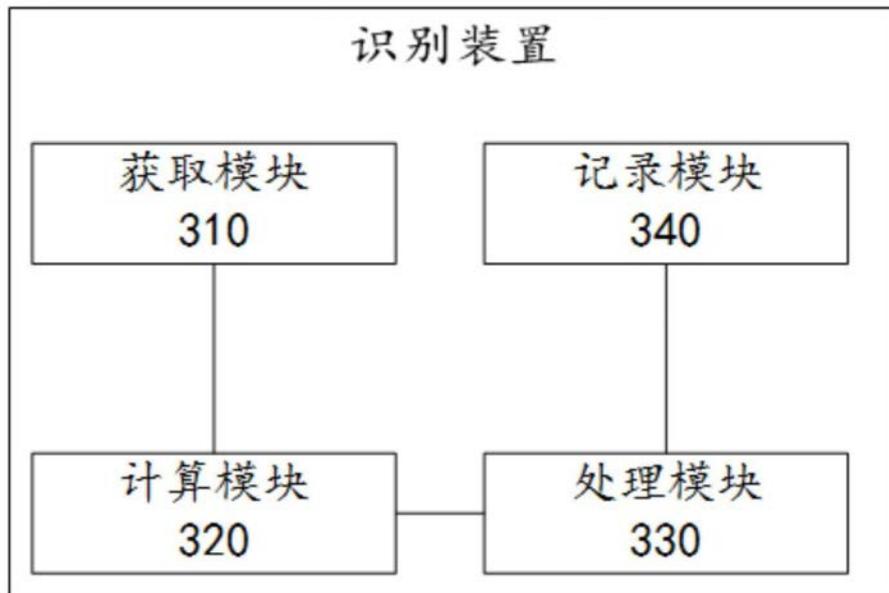


图3

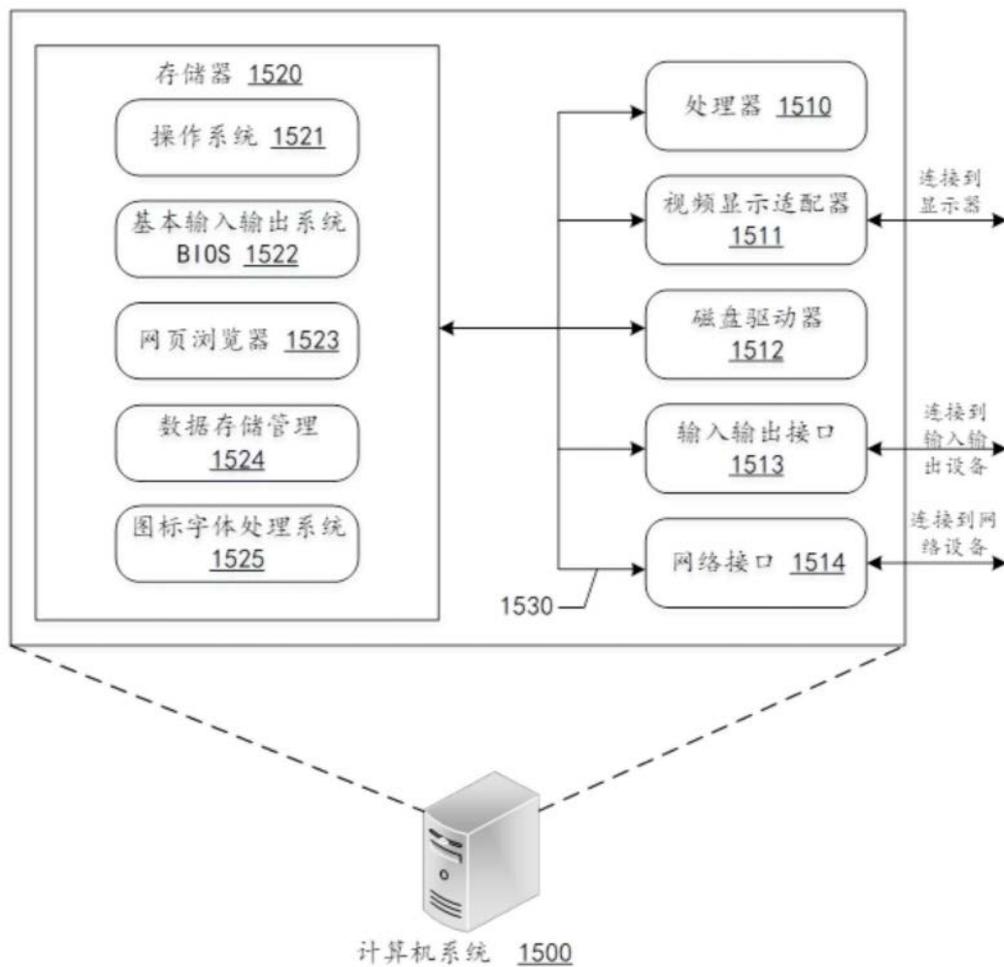


图4