



(12) 发明专利

(10) 授权公告号 CN 111045891 B

(45) 授权公告日 2024.03.19

(21) 申请号 201911267533.5

(22) 申请日 2019.12.11

(65) 同一申请的已公布的文献号

申请公布号 CN 111045891 A

(43) 申请公布日 2020.04.21

(73) 专利权人 深圳前海环融联易信息科技服务有限公司

地址 518000 广东省深圳市前海深港合作区前湾一路1号A栋201室(入驻深圳市前海商务秘书有限公司)

(72) 发明人 刘伟彬 肖隆韬 李荣

(74) 专利代理机构 深圳市精英专利事务所
44242

专利代理师 武志峰

(51) Int.Cl.

G06F 11/30 (2006.01)

(56) 对比文件

CN 103838654 A, 2014.06.04

CN 109284227 A, 2019.01.29

CN 110290212 A, 2019.09.27

CN 109766241 A, 2019.05.17

US 2018189160 A1, 2018.07.05

CN 110445669 A, 2019.11.12

US 2013227528 A1, 2013.08.29

审查员 陆海天

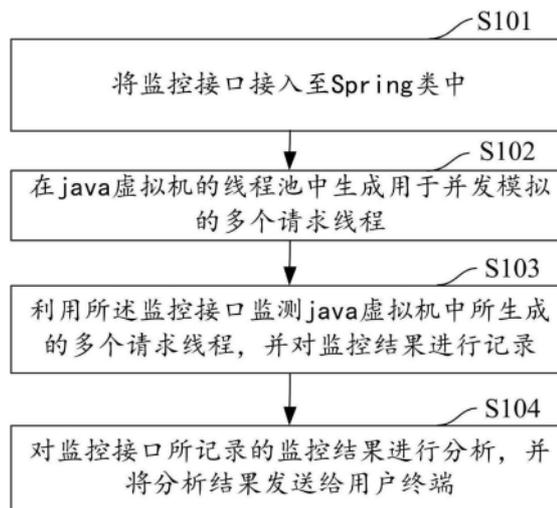
权利要求书2页 说明书7页 附图3页

(54) 发明名称

基于java多线程的监控方法、装置、设备以及存储介质

(57) 摘要

本发明实施例公开了一种基于java多线程的监控方法、装置、设备以及存储介质,所述方法包括:将监控接口接入至Spring类中;在java虚拟机的线程池中生成用于并发模拟的多个请求线程;利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录;对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端。本发明通过设置监控接口对并发线程的结果进行自动排查和分析,并将分析结果反馈给用户,能够节省大量的人力以及提高测试效率。



1. 一种基于java多线程的监控方法,其特征在于,所述方法包括:
在Spring类中声明一个切面类,并在所述切面类上使用第一预设标记进行标注;
在所述切面类中声明一个监控方法,并在该监控方法中注入监控代码;
对所注入的监控代码使用第二预设标记进行标注,并对所述监控代码进行编译;
在java虚拟机的线程池中生成用于并发模拟的多个请求线程;
利用监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录;
对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端;
其中,所述在java虚拟机的线程池中生成用于并发模拟的多个请求线程的步骤,包括:
通过继承线程池的Thread类生成用于并发模拟的多个请求线程;或者,
通过实现线程池的Runnable接口生成用于并发模拟的多个请求线程,其中,多个所述请求线程中的第一个请求线程用于获取java虚拟机的初始结果和初始性能参数,多个所述请求线程中其余请求线程用于监测java虚拟机的性能。

2. 如权利要求1所述的方法,其特征在于,所述利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录,包括:

利用所述监控接口记录请求线程运行时所获取的初始化性能参数、线程不安全行为信息以及线程的开销信息。

3. 如权利要求1所述的方法,其特征在于,所述对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端,包括:

将监控接口所记录的监控结果与预设的正常监控结果进行比较,若监控接口所记录的监控结果与预设的正常监控结果不一致,产生告警信息,并将告警信息发送给用户终端。

4. 一种基于java多线程的监控装置,其特征在于,所述装置包括:

第一标注单元,用于在Spring类中声明一个切面类,并在所述切面类上使用第一预设标记进行标注;

代码注入单元,用于在所述切面类中声明一个监控方法,并在该监控方法中注入监控代码;

第二标注单元,用于对所注入的监控代码使用第二预设标记进行标注,并对所述监控代码进行编译;

生成单元,用于在java虚拟机的线程池中生成用于并发模拟的多个请求线程;

监测记录单元,用于利用监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录;

分析发送单元,用于对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端;

其中,所述生成单元,具体用于通过继承线程池的Thread类生成用于并发模拟的多个请求线程;或者,

通过实现线程池的Runnable接口生成用于并发模拟的多个请求线程,其中,多个所述请求线程中的第一个请求线程用于获取java虚拟机的初始结果和初始性能参数,多个所述请求线程中其余请求线程用于监测java虚拟机的性能。

5. 一种计算机设备,包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现如权利要求1-3任

一项所述的基于 java 多线程的监控方法。

6. 一种计算机可读存储介质,其特征在於,所述计算机可读存储介质存储有一个或者一个以上计算机程序,所述一个或者一个以上计算机程序可被一个或者一个以上的处理器执行,以实现如权利要求1-3任一项所述的基于 java 多线程的监控方法。

基于java多线程的监控方法、装置、设备以及存储介质

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种基于java多线程的监控方法、装置、设备以及存储介质。

背景技术

[0002] 现有技术中,在一些并发场景下,我们通常对线程不安全的代码使用code review的方式进行排查。但是如果业务量较大,代码设计复杂,就难以通过code review的方式对线程不安全的代码进行排查,与此同时,我们还需要对线程的不安全问题进行性能分析,面对这种情形,如果仅通过人工的方式进行排查和分析,由于人的主观性在排查和分析的过程中很容易犯错,并且也会消耗比较大的人力。

发明内容

[0003] 有鉴于此,本发明实施例提供一种基于java多线程的监控方法、装置、设备以及存储介质,通过设置监控接口对并发线程的结果进行自动排查和分析,并将分析结果反馈给用户,能够节省大量的人力以及提高测试效率。

[0004] 一方面,本发明实施例提供了一种基于java多线程的监控方法,该方法包括:

[0005] 将监控接口接入至Spring类中;

[0006] 在java虚拟机的线程池中生成用于并发模拟的多个请求线程;

[0007] 利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录;

[0008] 对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端。

[0009] 另一方面,本发明实施例提供了一种基于java多线程的监控装置,所述装置包括:

[0010] 接入单元,用于将监控接口接入至Spring类中;

[0011] 生成单元,用于在java虚拟机的线程池中生成用于并发模拟的多个请求线程;

[0012] 监测记录单元,用于利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录;

[0013] 分析发送单元,用于对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端。

[0014] 又一方面,本发明实施例还提供了一种计算机设备,包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现如上所述的基于java多线程的监控方法。

[0015] 再一方面,本发明实施例还提供了一种计算机可读存储介质,所述计算机可读存储介质存储有一个或者一个以上计算机程序,所述一个或者一个以上计算机程序可被一个或者一个以上的处理器执行,以实现如上所述的基于java多线程的监控方法。

[0016] 本发明实施例提供一种基于java多线程的监控方法、装置、设备以及存储介质,其中方法包括:将监控接口接入至Spring类中;在java虚拟机的线程池中生成用于并发模拟

的多个请求线程;利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录;对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端。本发明通过设置监控接口对并发线程的结果进行自动排查和分析,并将分析结果反馈给用户,能够节省大量的人力以及提高测试效率。

附图说明

[0017] 为了更清楚地说明本发明实施例技术方案,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。

[0018] 图1是本发明实施例提供的一种基于java多线程的监控方法的应用场景示意图;

[0019] 图2是本发明实施例提供的一种基于java多线程的监控方法的示意流程图;

[0020] 图3是本发明实施例提供的一种基于java多线程的监控方法的另一示意流程图;

[0021] 图4是本发明实施例提供的一种基于java多线程的监控装置的示意性框图;

[0022] 图5是本发明实施例提供的一种基于java多线程的监控装置的另一示意性框图;

[0023] 图6是本发明实施例提供的一种计算机设备的结构组成示意图。

具体实施方式

[0024] 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。

[0025] 应当理解,当在本说明书和所附权利要求书中使用时,术语“包括”和“包含”指示所描述特征、整体、步骤、操作、元素和/或组件的存在,但并不排除一个或多个其它特征、整体、步骤、操作、元素、组件和/或其集合的存在或添加。

[0026] 还应当理解,在此本发明说明书中所使用的术语仅仅是出于描述特定实施例的目的而并不意在限制本发明。如在本发明说明书和所附权利要求书中所使用的那样,除非上下文清楚地指明其它情况,否则单数形式的“一”、“一个”及“该”意在包括复数形式。

[0027] 还应当进一步理解,在本发明说明书和所附权利要求书中使用的术语“和/或”是指相关联列出的项中的一个或多个的任何组合以及所有可能组合,并且包括这些组合。

[0028] 请参阅图1和图2,图1为本发明实施例提供的一种基于java多线程的监控方法的应用场景示意图,图2为本发明实施例提供的一种基于java多线程的监控方法的流程示意图。该基于java多线程的监控方法应用于java虚拟机中。作为一应用,如图1所示,该基于java多线程的监控方法应用于java虚拟机10中,该java虚拟机10通过监控和分析并发线程的运行结果,将分析结果发送给用户终端20,以通知用户例如开发人员根据所述分析结果进行后续处理,其中,用户终端为开发人员所使用的终端设备,例如台式电脑、笔记本或智能手机等等。

[0029] 请参阅图2,图2为本发明实施例提供的一种基于java多线程的监控方法的示意流程图。如图2所示,该方法包括以下步骤S101~S104。

[0030] S101,将监控接口接入至Spring类中。

[0031] 在本发明实施例中,所述监控接口由用户在Spring类中自定义编写监控代码编译形成,本实施例中通过aop(Aspect Oriented Programming,面向切面编程)的方式实现该监控接口,所谓aop指的是通过动态代理的方式,在方法执行前后或者抛出异常时进行拦截的一种编程方案,通过利用aop技术在动态代理的拦截器里编写监控目标方法的代码可形成监控接口。

[0032] 具体的,请参阅图3,所述步骤S101包括步骤S201~S203:

[0033] S201,在Spring类中声明一个切面类,并在所述切面类上使用第一预设标记进行标注。

[0034] 在本发明实施例中,通过AspectJ框架提供的切面功能,在Spring类中声明一个切面类,并在所声明的切面类上使用第一预设标记进行标注,例如使用“@Aspect”进行标注,标注的目的是为了说明这是一个切面类。

[0035] S202,在所述切面类中声明一个监控方法,并在该监控方法中注入监控代码。

[0036] 在本发明实施例中,在声明切面类之后,再在所声明的切面类中声明一个监控方法,并通过该监控方法编写所需监控的代码,在本实施例中,所编写的监控代码是面向请求线程的代码,具体的监控代码可以根据具体的实际应用场景进行选择,在此不作限制。

[0037] S203,对所注入的监控代码使用第二预设标记进行标注,并对所述监控代码进行编译。

[0038] 在本发明实施例中,在编写完监控代码之后,在该监控方法上使用第二预设标记进行标注,例如使用“@Around”进行标注,使用@Around进行标注的具体格式如下:@Around (“@annotation(自定义注解名)”),其中,注解名可以由用户自行选择,在此不作限制,在对监控代码进行标注后,对自定义注解进行编译,这样就将该监控接口接入到spring类中。

[0039] S102,在java虚拟机的线程池中生成用于并发模拟的多个请求线程。

[0040] 在本发明实施例中,所述在java虚拟机的线程池中生成用于并发模拟的多个请求线程具体包括:通过继承线程池的Thread类生成用于并发模拟的多个请求线程;或者,通过实现线程池的Runnable接口生成用于并发模拟的多个请求线程。具体的,通过在java虚拟机的线程池中生成用于并发模拟的多个请求线程,所生成的多个请求线程用于监测java虚拟机的性能,并通过运行获取java虚拟机的各项性能参数。在线程池中生成请求线程的方法可以通过继承线程池的Thread类得到,也可以通过实现线程池的Runnable接口的方式得到。其中,在线程池中所生成的第一个请求线程用于获取java虚拟机的初始结果和初始性能参数;当java虚拟机需要处理大量任务时,再在线程池中生成更多的请求线程以监测java虚拟机的性能。

[0041] S103,利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录。

[0042] 在本发明实施例中,所述利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录具体包括:利用所述监控接口记录请求线程运行时所获取的初始化性能参数、线程不安全行为信息以及线程的开销信息。具体的,线程池所生成的第一条请求线程,用于获取java虚拟机的初始结果和初始性能参数。当java虚拟机需要处理大量任务时,再在线程池中生成更多的请求线程以监测java虚拟机的性能。再利用所述监控接口记录请求线程运行时所获取的初始化性能参数、线程不安全行为信息以及线程的开销

信息。

[0043] S104,对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端。

[0044] 在本发明实施例中,所述对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端具体包括:将监控接口所记录的监控结果与预设的正常监控结果进行比较,若监控接口所记录的监控结果与预设的正常监控结果不一致,产生告警信息,并将告警信息发送预设的用户终端以通知相关用户终端。

[0045] 具体的,在并发模拟的请求线程运行结束后,需要对监控接口的监控结果的进行分析。在分析监控结果之前,预先准备好正常的监控结果以备对比分析。若监控接口所记录的监控结果与预设的正常监控结果不一致,那么可以确定被监控的线程存在线程安全的问题。例如:在并发模拟多请求线程时,若有栈溢出可能的原因是递归深度太大,若响应时间过长可能的原因是循环查询。通过对结果的分析,并反馈给开发人员,具体可以通过邮件的形式通知开发人员对存在问题的线程进行优化,通过监控接口实现自动排查和分析,并将分析结果反馈给开发人员,能够节省大量的人力以及提高测试效率。

[0046] 由以上可见,本发明实施例通过将监控接口接入至Spring类中;在java虚拟机的线程池中生成用于并发模拟的多个请求线程;利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录;对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端。本发明通过设置监控接口对并发线程的结果进行自动排查和分析,并将分析结果反馈给用户,能够节省大量的人力以及提高测试效率。

[0047] 请参阅图4,对应上述一种基于java多线程的监控方法,本发明实施例还提出一种基于java多线程的监控装置,该装置100包括:接入单元101、生成单元102、监测记录单元103、分析发送单元104。

[0048] 其中,所述接入单元101,用于将监控接口接入至Spring类中;生成单元102,用于在java虚拟机的线程池中生成用于并发模拟的多个请求线程;监测记录单元103,用于利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录;分析发送单元104,用于对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端。

[0049] 请参阅图5,所述接入单元101,包括:第一标注单元101a,用于在Spring类中声明一个切面类,并在所述切面类上使用第一预设标记进行标注;代码注入单元101b,用于在所述切面类中声明一个监控方法,并在该监控方法中注入监控代码;第二标注单元101c,用于对所注入的监控代码使用第二预设标记进行标注,并对所述监控代码进行编译。

[0050] 所述生成单元102,具体用于通过继承线程池的Thread类生成用于并发模拟的多个请求线程;或者,通过实现线程池的Runnable接口生成用于并发模拟的多个请求线程。

[0051] 所述监测记录单元103,具体用于利用所述监控接口记录请求线程运行时所获取的初始化性能参数、线程不安全行为信息以及线程的开销信息。

[0052] 所述分析发送单元104,具体用于将监控接口所记录的监控结果与预设的正常监控结果进行比较,若监控接口所记录的监控结果与预设的正常监控结果不一致,产生告警信息,并将告警信息发送给用户终端。

[0053] 由以上可见,本发明实施例通过将监控接口接入至Spring类中;在java虚拟机的线程池中生成用于并发模拟的多个请求线程;利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录;对监控接口所记录的监控结果进行分析,并将分

析结果发送给用户终端。本发明通过设置监控接口对并发线程的结果进行自动排查和分析,并将分析结果反馈给用户,能够节省大量的人力以及提高测试效率。

[0054] 上述基于java多线程的监控装置与上述基于java多线程的监控方法一一对应,其具体的原理和过程与上述实施例所述方法相同,不再赘述。

[0055] 上述基于java多线程的监控装置可以实现为一种计算机程序的形式,计算机程序可以在如图6所示的计算机设备上运行。

[0056] 图6为本发明一种计算机设备的结构组成示意图。该设备可以是终端,也可以是服务器,其中,终端可以是笔记本电脑以及台式电脑等电子装置。服务器可以是独立的服务器,也可以是多个服务器组成的服务器集群。参照图6,该计算机设备500包括通过系统总线501连接的处理器502、非易失性存储介质503、内存储器504和网络接口505。其中,该计算机设备500的非易失性存储介质503可存储操作系统5031和计算机程序5032,该计算机程序5032被执行时,可使得处理器502执行一种基于java多线程的监控方法。该计算机设备500的处理器502用于提供计算和控制能力,支撑整个计算机设备500的运行。该内存储器504为非易失性存储介质503中的计算机程序5032的运行提供环境,该计算机程序被处理器执行时,可使得处理器502执行一种基于java多线程的监控方法。计算机设备500的网络接口505用于进行网络通信。本领域技术人员可以理解,图6中所示仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的计算机设备的限定,具体的计算机设备可以包括比图中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。

[0057] 其中,所述处理器502执行所述计算机程序时实现如下操作:

[0058] 将监控接口接入至Spring类中;

[0059] 在java虚拟机的线程池中生成用于并发模拟的多个请求线程;

[0060] 利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录;

[0061] 对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端。

[0062] 在一个实施例中,所述将监控接口接入至Spring类中,包括:

[0063] 在Spring类中声明一个切面类,并在所述切面类上使用第一预设标记进行标注;

[0064] 在所述切面类中声明一个监控方法,并在该监控方法中注入监控代码;

[0065] 对所注入的监控代码使用第二预设标记进行标注,并对所述监控代码进行编译。

[0066] 在一个实施例中,所述在java虚拟机的线程池中生成用于并发模拟的多个请求线程,包括:

[0067] 通过继承线程池的Thread类生成用于并发模拟的多个请求线程;或者,通过实现线程池的Runnable接口生成用于并发模拟的多个请求线程。

[0068] 在一个实施例中,所述利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录,包括:

[0069] 利用所述监控接口记录请求线程运行时所获取的初始化性能参数、线程不安全行为信息以及线程的开销信息。

[0070] 在一个实施例中,所述对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端,包括:

[0071] 将监控接口所记录的监控结果与预设的正常监控结果进行比较,若监控接口所记

录的监控结果与预设的正常监控结果不一致,产生告警信息,并将告警信息发送给用户终端。

[0072] 本领域技术人员可以理解,图6中示出的计算机设备的实施例并不构成对计算机设备具体构成的限定,在其他实施例中,计算机设备可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。例如,在一些实施例中,计算机设备仅包括存储器及处理器,在这样的实施例中,存储器及处理器的结构及功能与图6所示实施例一致,在此不再赘述。

[0073] 本发明提供了一种计算机可读存储介质,计算机可读存储介质存储有一个或者一个以上计算机程序,所述一个或者一个以上计算机程序可被一个或者一个以上的处理器执行,以实现以下步骤:

[0074] 将监控接口接入至Spring类中;

[0075] 在java虚拟机的线程池中生成用于并发模拟的多个请求线程;

[0076] 利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录;

[0077] 对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端。

[0078] 在一个实施例中,所述将监控接口接入至Spring类中,包括:

[0079] 在Spring类中声明一个切面类,并在所述切面类上使用第一预设标记进行标注;

[0080] 在所述切面类中声明一个监控方法,并在该监控方法中注入监控代码;

[0081] 对所注入的监控代码使用第二预设标记进行标注,并对所述监控代码进行编译。

[0082] 在一个实施例中,所述在java虚拟机的线程池中生成用于并发模拟的多个请求线程,包括:

[0083] 通过继承线程池的Thread类生成用于并发模拟的多个请求线程;或者,通过实现线程池的Runnable接口生成用于并发模拟的多个请求线程。

[0084] 在一个实施例中,所述利用所述监控接口监测java虚拟机中所生成的多个请求线程,并对监控结果进行记录,包括:

[0085] 利用所述监控接口记录请求线程运行时所获取的初始化性能参数、线程不安全行为信息以及线程的开销信息。

[0086] 在一个实施例中,所述对监控接口所记录的监控结果进行分析,并将分析结果发送给用户终端,包括:

[0087] 将监控接口所记录的监控结果与预设的正常监控结果进行比较,若监控接口所记录的监控结果与预设的正常监控结果不一致,产生告警信息,并将告警信息发送给用户终端。

[0088] 本发明前述的存储介质包括:磁碟、光盘、只读存储记忆体(Read-Only Memory, ROM)等各种可以存储程序代码的介质。

[0089] 本发明所有实施例中的单元可以通过通用集成电路,例如CPU(Central Processing Unit,中央处理器),或通过ASIC(Application Specific Integrated Circuit,专用集成电路)来实现。

[0090] 本发明实施例基于java多线程的监控方法中的步骤可以根据实际需要进行顺序调整、合并和删减。

[0091] 本发明实施例基于 java 多线程的监控装置中的单元可以根据实际需要进行合并、划分和删减。

[0092] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到各种等效的修改或替换,这些修改或替换都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以权利要求要求的保护范围为准。

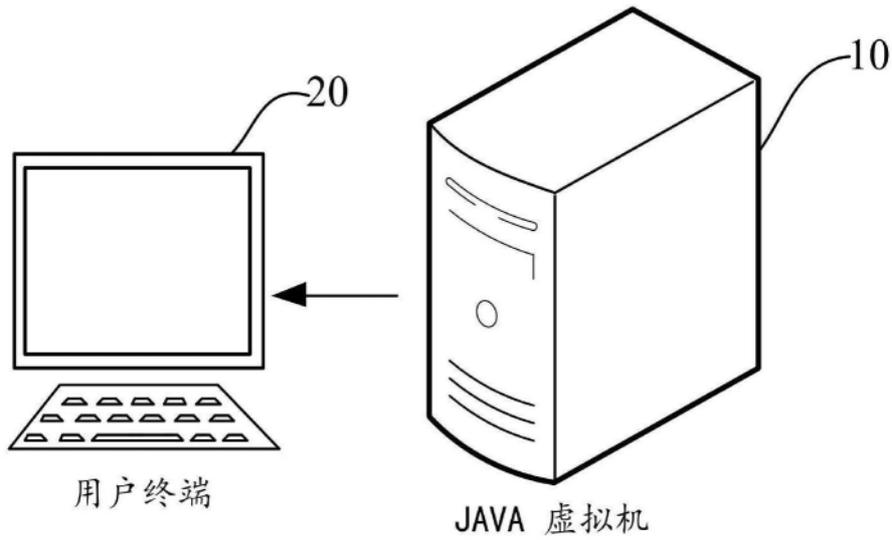


图1

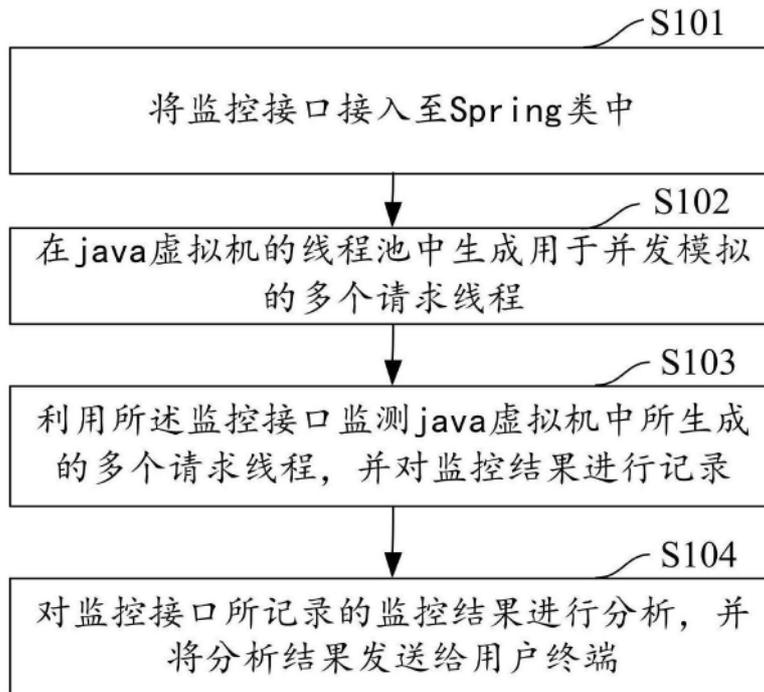


图2

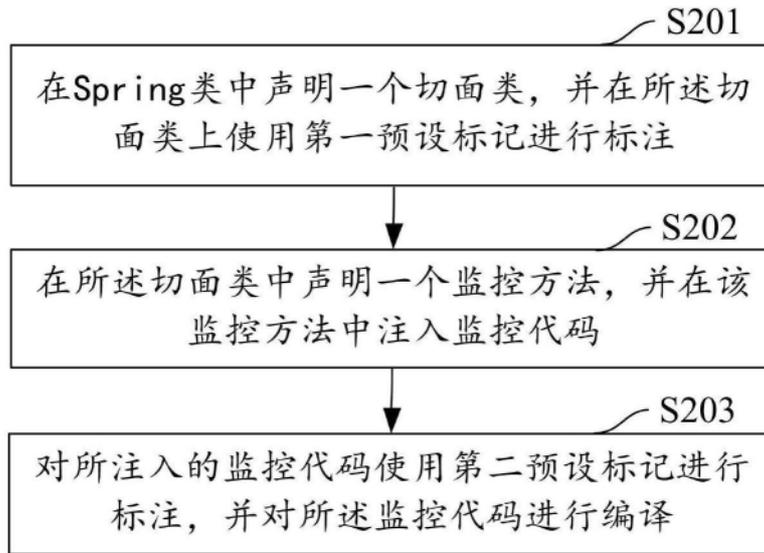


图3

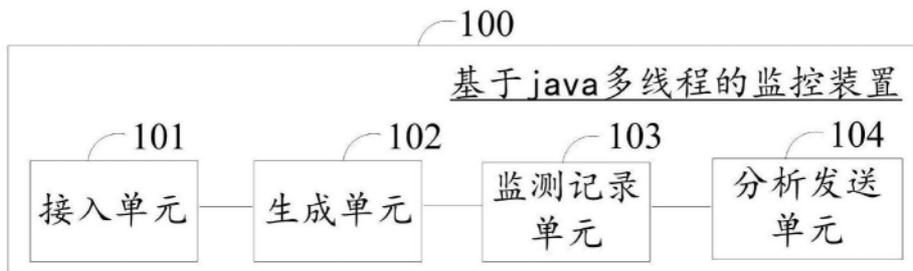


图4

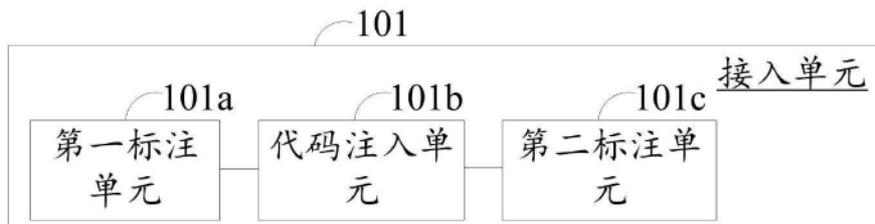


图5

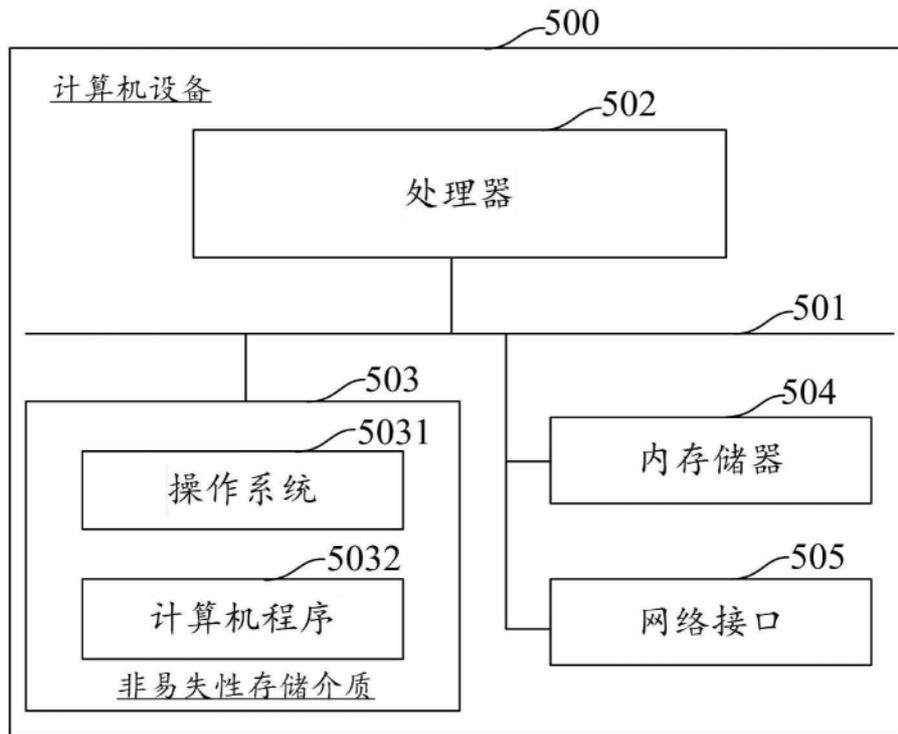


图6