(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 113238815 A
(43)申请公布日 2021.08.10

(21)申请号 202110524425.2

(22)申请日 2021.05.13

(71)申请人 北京京东振世信息技术有限公司
地址 100086 北京市海淀区知春路76号6层

(72)发明人 吴贻淮 姚古斌 骆彬彬 胡雄 孙向前

(74)专利代理机构 北京品源专利代理有限公司 11332
代理人 孟金喆

(51)Int.Cl.
G06F 9/445(2018.01)

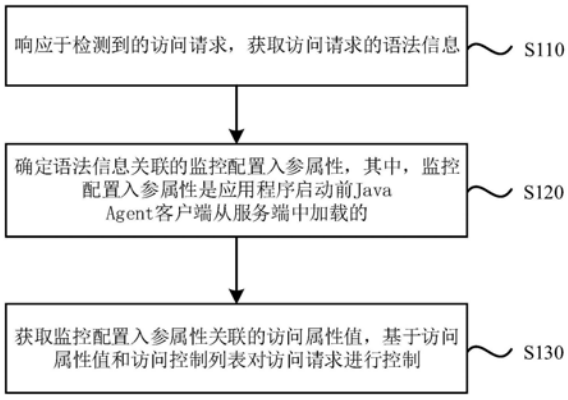
权利要求书2页 说明书13页 附图6页

(54)发明名称

一种接口访问控制方法、装置、设备及存储介质

(57)摘要

本发明实施例公开了一种接口访问控制方法、装置、设备及存储介质，所述方法包括：响应于检测到的访问请求，获取所述访问请求的语法信息；确定所述语法信息关联的监控配置入参属性，其中，所述监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的；获取所述监控配置入参属性关联的访问属性值，基于所述访问属性值和访问控制列表对所述访问请求进行控制。本发明实施例提供的方法通过预先配置访问控制信息，在应用程序启动前从服务端获取并加载，使得接口的访问控制配置更加方便，减少了Java应用的接入复杂度，使得Java应用的大量接入得以实现。



CN 113238815 A

1.一种接口访问控制方法,其特征在于,包括:

响应于检测到的访问请求,获取所述访问请求的语法信息;

确定所述语法信息关联的监控配置入参属性,其中,所述监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的;

获取所述监控配置入参属性关联的访问属性值,基于所述访问属性值和访问控制列表对所述访问请求进行控制。

2.根据权利要求1所述的方法,其特征在于,所述基于所述访问属性值和访问控制列表对所述访问请求进行控制,包括:

将所述访问属性值与所述访问控制列表中的控制属性值进行匹配;

当所述访问属性值与所述控制属性值匹配成功时,根据预先配置的回调结果进行降级熔断,生成访问失败的响应信息进行展示;

当所述访问属性值与所述控制属性值未匹配成功时,执行所述访问请求。

3.根据权利要求1所述的方法,其特征在于,还包括:

将所述访问属性值发送至服务端,以使服务端生成所述访问属性值关联的访问分析结果;

接收服务端发送的访问分析结果,基于所述访问分析结果对所述访问控制列表进行更新。

4.根据权利要求1所述的方法,其特征在于,还包括:

应用程序启动前,Java Agent客户端根据所述客户端关联的应用标识从服务端中加载预先配置的访问控制配置信息;

基于所述访问控制配置信息对加载的Java类进行逻辑入侵,完成Java类的加载。

5.一种接口访问控制方法,其特征在于,包括:

接收客户端发送的访问控制配置信息获取请求;

根据所述访问控制配置信息获取请求对应的应用标识确定目标访问控制配置信息;

基于所述目标访问控制配置信息生成获取请求响应信息发送至客户端。

6.根据权利要求5所述的方法,其特征在于,还包括:

响应于检测到的访问控制信息配置请求,生成所述访问控制信息配置请求对应应用的访问控制信息配置界面并展示;

响应于检测到的控制信息配置完成请求,获取所述控制信息配置完成请求关联的访问控制信息,并将所述访问控制信息与应用标识关联存储。

7.一种接口访问控制装置,其特征在于,包括:

语法信息获取模块,用于响应于检测到的访问请求,获取所述访问请求的语法信息;

监控入参获取模块,用于确定所述语法信息关联的监控配置入参属性,其中,所述监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的;

访问请求控制模块,用于获取所述监控配置入参属性关联的访问属性值,基于所述访问属性值和访问控制列表对所述访问请求进行控制。

8.一种接口访问控制装置,其特征在于,包括:

信息获取请求模块,用于接收客户端发送的访问控制配置信息获取请求;

目标配置信息模块,用于根据所述访问控制配置信息获取请求对应的应用标识确定目

标访问控制配置信息；

响应信息生成模块，用于基于所述目标访问控制配置信息生成获取请求响应信息发送至客户端。

9.一种计算机设备，其特征在于，所述设备包括：

一个或多个处理器；

存储装置，用于存储一个或多个程序；

当所述一个或多个程序被所述一个或多个处理器执行，使得所述一个或多个处理器实现如权利要求1-4中任一所述的接口访问控制方法，和/或，实现如权利要求5或6所述的接口访问控制方法。

10.一种计算机可读存储介质，其上存储有计算机程序，其特征在于，该程序被处理器执行时实现如权利要求1-4中任一所述的接口访问控制方法，和/或，实现如权利要求5或6所述的接口访问控制方法。

一种接口访问控制方法、装置、设备及存储介质

技术领域

[0001] 本发明实施例涉及计算机技术领域，尤其涉及一种接口访问控制方法、装置、设备及存储介质。

背景技术

[0002] 随着计算机微服务技术的高速发展，Java技术被广泛应用于各大分布式系统的服务端，因而Java微服务接口的防刷、限流也一直被是一个备受关注的技术问题，目前开源的Java微服务接口防刷、限流技术有Hystrix、Sentinel等。在实现本发明的过程中，发明人发现现有技术中至少存在以下技术问题：现有开源的Hystrix、Sentinel客户端采用注解形式接入，意味着任何需要接入的Java应用都需要进行一定量的代码改造，接入过程复杂，不方便于的Java用的大量接入。

发明内容

[0003] 本发明实施例提供了一种接口访问控制方法、装置、设备及存储介质，以实现简化Java应用接入的复杂度。

[0004] 第一方面，本发明实施例提供了一种接口访问控制方法，应用于客户端，包括：

[0005] 响应于检测到的访问请求，获取访问请求的语法信息；

[0006] 确定语法信息关联的监控配置入参属性，其中，监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的；

[0007] 获取监控配置入参属性关联的访问属性值，基于访问属性值和访问控制列表对访问请求进行控制。

[0008] 第二方面，本发明实施例还提供了一种接口访问控制方法，应用于服务端，包括：

[0009] 接收客户端发送的访问控制配置信息获取请求；

[0010] 根据访问控制配置信息获取请求对应的应用标识确定目标访问控制配置信息；

[0011] 基于目标访问控制配置信息生成获取请求响应信息发送至客户端。

[0012] 第三方面，本发明实施例还提供了一种接口访问控制装置，配置于客户端，包括：

[0013] 语法信息获取模块，用于响应于检测到的访问请求，获取访问请求的语法信息；

[0014] 监控入参获取模块，用于确定语法信息关联的监控配置入参属性，其中，监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的；

[0015] 访问请求控制模块，用于获取监控配置入参属性关联的访问属性值，基于访问属性值和访问控制列表对访问请求进行控制。

[0016] 第四方面，本发明实施例还提供了一种接口访问控制装置，配置于服务端，包括：

[0017] 信息获取请求模块，用于接收客户端发送的访问控制配置信息获取请求；

[0018] 目标配置信息模块，用于根据访问控制配置信息获取请求对应的应用标识确定目标访问控制配置信息；

[0019] 响应信息生成模块，用于基于目标访问控制配置信息生成获取请求响应信息发送

至客户端。

[0020] 第五方面，本发明实施例还提供了一种计算机设备，设备包括：

[0021] 一个或多个处理器；

[0022] 存储装置，用于存储一个或多个程序；

[0023] 当一个或多个程序被一个或多个处理器执行，使得一个或多个处理器实现如本发明实施例第一方面所提供的接口访问控制方法，和/或，实现如本发明实施例第二方面所提供的接口访问控制方法。

[0024] 第六方面，本发明实施例还提供了一种计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时实现如本发明实施例第一方面所提供的接口访问控制方法，和/或，实现如本发明实施例第二方面所提供的接口访问控制方法。

[0025] 本发明实施例通过响应于检测到的访问请求，获取访问请求的语法信息；确定语法信息关联的监控配置入参属性，其中，监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的；获取监控配置入参属性关联的访问属性值，基于访问属性值和访问控制列表对访问请求进行控制，通过预先配置访问控制信息，在应用程序启动前从服务端获取并加载，使得接口的访问控制配置更加方便，减少了Java应用的接入复杂度，使得Java应用的大量接入得以实现。

附图说明

[0026] 图1是本发明实施例一所提供的一种接口访问控制方法的流程图；

[0027] 图2是本发明实施例二所提供的一种接口访问控制方法的流程图；

[0028] 图3a是本发明实施例二所提供的一种接口访问控制系统的结构图；

[0029] 图3b是本发明实施例三所提供的一种监控点配置和客户端启动时序图；

[0030] 图3c是本发明实施例三所提供的一种监控点调用流程示意图；

[0031] 图4是本发明实施例四所提供的一种接口访问控制装置的结构示意图；

[0032] 图5是本发明实施例五所提供的一种接口访问控制装置的结构示意图；

[0033] 图6是本发明实施例六所提供的一种计算机设备的结构示意图。

具体实施方式

[0034] 下面结合附图和实施例对本发明作进一步的详细说明。可以理解的是，此处所描述的具体实施例仅仅用于解释本发明，而非对本发明的限定。另外还需要说明的是，为了便于描述，附图中仅示出了与本发明相关的部分而非全部结构。

[0035] 实施例一

[0036] 图1是本发明实施例一所提供的一种接口访问控制方法的流程图。本实施例可适用于对客户端的访问进行控制时的情形。该方法可以由接口访问控制装置执行，该接口访问控制装置可以采用软件和/或硬件的方式实现，例如，该接口访问控制装置可配置于计算机设备(如客户端)中。如图1所示，该方法包括：

[0037] S110、响应于检测到的访问请求，获取访问请求的语法信息。

[0038] 在本实施例中，访问请求为用户通过客户端发起的，用于获取指定信息的请求。示例性的，假设用户需要查看物品A的详情信息，可以通过点击物品A触发访问请求，客户端即

检测到用户发起的访问请求。

[0039] 一般的，监控点是用户基于语法信息进行配置的，如对某个类下的某个方法进行监控，则将该类下的该方法设置为监控点。因此，在检测到用户发起的访问请求后，获取访问请求的语法信息。其中，访问请求的语法信息可以包括访问请求中的类和/或方法信息。

[0040] S120、确定语法信息关联的监控配置入参属性，其中，监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的。

[0041] 可选的，确定访问请求关联的语法信息后，根据应用程序启动前加载的访问控制配置信息获取语法信息关联的监控配置入参属性。示例性的，入参属性可以为用户标识、设备标识等属性。

[0042] 获取访问请求的语法信息后，从访问控制配置信息中查找与语法信息对应的配置入参标识，将其作为语法信息关联的监控配置入参属性。示例性的，假设访问控制以用户为控制单位，则可以将监控配置入参属性设置为用户标识。

[0043] 在本发明的一种实施方式中，还包括：应用程序启动前，Java Agent客户端根据客户端关联的应用标识从服务端中加载预先配置的访问控制配置信息；基于访问控制配置信息对加载的Java类进行逻辑入侵，将完成逻辑入侵后的新Java类，推送至Java虚拟机进行正常的类加载，完成Java类的加载。本实施例在服务端进行访问控制的配置，在应用程序启动前，先通过Java Agent客户端到通过应用标识向远程服务发起访问控制配置信息获取请求，接收服务端返回的响应信息(即访问控制配置信息)，根据访问控制配置信息中的监控点信息，对加载的Java类进行逻辑入侵，完成Java类的加载。在将完成逻辑入侵后的新Java类，交给Java虚拟机进行正常的类加载后，即可接收用户发起的访问请求。可选的，在完成Java类的加载后，客户端还可以启动监控点配置信息定时拉取任务以及监控点统计信息定时上报任务，实现对监控点的监控以及监控点信息的统计分析。

[0044] S130、获取监控配置入参属性关联的访问属性值，基于访问属性值和访问控制列表对访问请求进行控制。

[0045] 在本实施例中，确定监控配置入参属性后，将访问信息中监控配置入参属性对应的值作为访问属性值，基于访问属性值和本地存储的访问控制列表判断是否需要对访问请求进行降级熔断等控制。示例性的，假设监控配置入参属性为用户标识，则获取访问请求中携带的用户标识作为访问属性值，判断访问控制列表中是否存在该访问属性值，根据判断结果确定是否需要对访问请求进行降级熔断等控制。

[0046] 一个实施例中，基于访问属性值和访问控制列表对访问请求进行控制，包括：将访问属性值与访问控制列表中的控制属性值进行匹配；当访问属性值与控制属性值匹配成功时，根据预先配置的回调结果进行降级熔断，生成访问失败的响应信息进行展示；当访问属性值与控制属性值未匹配成功时，执行访问请求。也就是说，判断本地的访问控制列表中是否存在该访问属性值，若存在，则根据访问控制配置信息所配置的回调结果进行降级熔断，如果不在则执行访问请求的原有逻辑。

[0047] 在上述方案的基础上，还包括：将访问属性值发送至服务端，以使服务端生成访问属性值关联的访问分析结果；接收服务端发送的访问分析结果，基于访问分析结果对访问控制列表进行更新。在对访问请求进行降级熔断或执行后，将访问请求的访问属性值发送至服务端，服务端根据接收到的访问属性值以及历史访问情况对该访问属性值进行分析，

得到该访问属性值的分析结果，发送至客户端，客户端基于服务端发送的访问分析结果更新访问控制列表。示例性的，对访问属性值进行分析可以为：判断访问属性值关联的访问次数是否超过访问控制配置信息中配置的访问阈值；访问分析结果可以为访问属性值是否添加至访问控制列表中。

[0048] 本发明实施例通过响应于检测到的访问请求，获取访问请求的语法信息；确定语法信息关联的监控配置入参属性，其中，监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的；获取监控配置入参属性关联的访问属性值，基于访问属性值和访问控制列表对访问请求进行控制，通过预先配置访问控制信息，在应用程序启动前从服务端获取并加载，使得接口的访问控制配置更加方便，减少了Java应用的接入复杂度，使得Java应用的大量接入得以实现。

[0049] 实施例二

[0050] 图2是本发明实施例二所提供的一种接口访问控制方法的流程图。本实施例可适用于对客户端的访问进行控制时的情形。该方法可以由接口访问控制装置执行，该接口访问控制装置可以采用软件和/或硬件的方式实现，例如，该接口访问控制装置可配置于计算机设备(如服务端)中。如图2所示，该方法包括：

[0051] S210、接收客户端发送的访问控制配置信息获取请求。

[0052] 在本实施例中，访问控制配置信息获取请求是应用程序启动前Java Agent客户端向服务端发起的。服务端在接收到访问控制配置信息获取请求后，向客户端下发访问控制配置信息获取请求对应的访问控制配置信息。

[0053] 可选的，在接收客户端发送的访问控制配置信息获取请求之前，还包括：响应于检测到的访问控制信息配置请求，生成访问控制信息配置请求对应应用的访问控制信息配置界面并展示；响应于检测到的控制信息配置完成请求，获取控制信息配置完成请求关联的访问控制信息，并将访问控制信息与应用标识关联存储。在本实施例中，通过用户在服务端配置访问控制信息，实现Java应用接入的访问控制。可选的，用户进入服务端系统创建页面创建系统，然后进入服务端应用创建页面给对应的系统创建应用，然后点击服务端监控点创建控件触发访问控制信息配置请求，进入服务端监控点创建页面给对应的应用创建监控点。配置信息可以包括：所属应用，监控点类路径，监控点方法名称，备注信息等，内容存储在监控点表中。最后用户进入服务端防刷限流策略创建页面给对应的应用监控点创建防刷限流策略。配置信息可以包括：所属监控点，策略名称，防刷限流策略统计的入参属性名称，周期内的调用阀值次数，监控的异常类型，监控的异常阀值，防刷限流回调类型，防刷限流回调结果等信息，内容存放在防刷限流策略表中。完成访问控制信息的配置及存储。

[0054] S220、根据访问控制配置信息获取请求对应的应用标识确定目标访问控制配置信息。

[0055] 服务端在接收到访问控制配置信息获取请求后，获取访问控制配置信息获取请求中携带的应用标识，将应用标识对应的访问控制配置信息作为目标访问控制信息。

[0056] S230、基于目标访问控制配置信息生成获取请求响应信息发送至客户端。

[0057] 在本实施例中，可以直接将目标访问控制配置信息作为响应信息发送至客户端，以使客户端根据接收到的响应信息对加载的Java类进行逻辑入侵，完成Java类的加载，实现通过加载的Java类对访问请求进行控制。

[0058] 本发明实施例通过接收客户端发送的访问控制配置信息获取请求;根据访问控制配置信息获取请求对应的应用标识确定目标访问控制配置信息;基于目标访问控制配置信息生成获取请求响应信息发送至客户端,通过预先配置访问控制信息,在应用程序启动前将Java Agent客户端到服务端获取的访问控制信息下发,使得接口的访问控制配置更加方便,减少了Java应用的接入复杂度,使得Java应用的大量接入得以实现。

[0059] 实施例三

[0060] 本实施例在上述方案的基础上,提供了一种优选实施例。本实施例中,将访问控制配置信息具体为防刷限流策略,对接口访问控制方法进行说明。

[0061] 图3a是本发明实施例二所提供的一种接口访问控制系统的结构图。本发明实施例三所提供的接口访问控制系统基于Java Agent的Java接口实现。如图3a所示,接口访问控制系统包括客户端和服务端,其中,客户端由监控点拉取模块,防刷限流策略执行模块,监控信息采集上报模块组成。服务端由配置中心、流量分析模块组成。

[0062] 具体的,配置中心用于存储监控点的配置信息,客户端通过网络请求定时从服务端拉取配置信息,并根据配置信息判断当前请求是否需要进行拦截降级操作,配置中心涉及的数据表有:系统信息表、应用信息表、用户系统表、监控点表、防刷限流策略表、心跳表。其表结构如下表1、表2、表3、表4、表5、表6所示。其中,系统信息表用于存储监控点所属的系统信息;应用信息表用于存储监控点所属的应用信息,应用表与系统表是多对一的关系;用户系统表用于存储系统与用户之间的对应关系;监控点表用于存储监控点信息;防刷限流策略表用于存储监控点的防刷限流策略与防刷接口的关系为多对一的关系;心跳表用于存储客户端各个机器上报的心跳信息方便服务端进行报表统计。

[0063] 表1系统信息表

[0064]

字段	字段含义	备注
Id	数据库表id	
system_name	系统名称	
Create_user_id	创建用户id	
Create_time	创建时间	
Update_user_id	更新时间	
Update_time	更新时间	
Is_delete	是否有效	
Remark	备注信息	

[0065] 表2应用信息表

[0066]

字段	字段含义	备注
Id	数据库表id	
System_id	所属的系统id	
App_name	应用名称	
Create_user_id	创建人id	
Create_time	创建时间	
Update_user_id	更新人id	
Update_time	更新时间	

Is_delete	是否被删除	
Remark	备注信息	

[0067] 表3用户系统表

[0068]

字段	字段含义	备注
Id	数据库表 id	

[0069]

System_id	所属的系统 id	
User_id	用户 id 表	
Is_delete	是否被删除	
Create_time	创建时间	

[0070] 表4监控点表

[0071]

字段	字段含义	备注
Id	数据库表id	
App_id	所属应用id	
Class_path	监控点类路径	
class_Method	监控点方法名称	
Create_user_id	创建用户id	
Create_time	创建时间	
Update_user_id	更新用户id	
Update_time	更新时间	
Is_delete	是否删除	
Remark	备注	

[0072] 表5防刷限流策略表

[0073]

字段	字段含义	备注
Id	数据库表 id	
Method_id	所属应用 id	
Strategy_name	策略名称	
Key_work	防刷限流策略统计的入参属性名称	例如 arg0.customId，表示监控接口的第 1 个参数的 customdId 属性
Request_times	周期内的调用阀值次数	
Request_exception_times	周期内的接口异常阀值次数	

[0074]

Request_exception_Type	异常类型	
CallBack_Type	防刷回调类型	
CallBack_Result	防刷限流回调结果	触发防刷策略后客户端需要执行的结果
Is_delete	是否删除	
Remark	备注	

[0075] 表6心跳表

[0076]

字段	字段含义	备注
Id	数据库表id	
Ip	客户端ip	
Method_id	接口id	
Requet_time	周期时间内的调用次数	
Exception_time	周期时间内的异常次数	
Callback_time	周期时间内的降级熔断次数	
Create_time	创建时间	
Update_time	更新时间	
Report_time	上报时间	

[0077] 流分析模块主要用于接收客户端上报的监控信息，并对客户端上报的监控属性值进行统计，并结合防刷限流策略表中的阀值信息判断当前关键字是否需要触发防刷限流回

调结果，并将分析结果返回客户端，客户端将在当前统计周期内信任客户端的分析结果，并将该值维护到本地临时列表中，该值只在当前统计周期内有效。

[0078] 监控点模块作为客户端的核心模块，在系统启动的类加载之前，先访问服务端，通过预先分配的应用id，拉取预先在服务端配置的监控点防刷限流策略。在类加载时如果当前加载类方法预先在服务端配置过防刷限流策略，则通过Java Agent技术进行逻辑入侵，入侵内容如下：

[0079] (1)记录统计周期内的调用次数；

[0080] (2)记录统计周期内的异常次数；

[0081] (3)执行防刷限流策略模块，判断当前请求是否需要进行降级熔断处理；

[0082] (4)异步上报请求中的监控点需要监控的属性值，以便服务端进行流量分析，并接受服务端返回分析结果，该结果只在规定周期(预先配置的防刷限流策略中配置)内有效。

[0083] 另外监控点拉取模块，需要按照固定频率，到服务端拉取最新的监控点防刷限流策略，以急时同步服务端的最新配置信息。

[0084] 防刷限流策略执行模块用于执行服务端配置的防刷限流策略。

[0085] 监控信息采集上报模块用于上报客户端自身的统计结果，统计内容包括周期内的监控点的调用次数、周期时间内监控点的异常次数，周期时间内监控点触发防刷限流次数等信息，以便服务端进行报表展示。

[0086] 图3b是本发明实施例三所提供的一种监控点配置和客户端启动时序图。如图3b所示，监控点配置创建过程包括：

[0087] (1)用户进入服务端系统创建页面创建系统。填写信息包括，系统名称和备注信息，内容存储在系统信息表中。

[0088] (2)服务端通过页面弹窗形式告知用户系统是否创建成功。

[0089] (3)用户进入服务端应用创建页面给对应的系统创建应用。填写信息包括，所示系统，应用名称，和备注信息，应用信息表中。

[0090] (4)服务端通过页面弹窗形式告知用户应用是否创建成功。

[0091] (5)用户进入服务端监控点创建页面给对应的应用创建监控点。填写内容包括，所属应用，监控点类路径，监控点方法名称，备注信息等，内容存储在监控点表中。

[0092] (6)服务端通过页面弹窗形式告知用户监控点是否创建成功。

[0093] (7)用户进入服务端防刷限流策略创建页面给对应的应用创建防刷限流策略。填写内容包括，所属监控点，策略名称，防刷限流策略统计的入参属性名称，周期内的调用阀值次数，监控的异常类型，监控的异常阀值，防刷限流回调类型，防刷限流回调结果等信息，内容存放在防刷限流策略表中。

[0094] (8)服务端通过页面弹窗形式告知用户防刷限流策略是否创建成功。

[0095] 上述步骤控点配置创建过程，用户可以通过服务端提供的页面完成所需接入应用的监控点配置创建。

[0096] 在应用程序启动前，从服务端获取预先配置的防刷限流策略表，具体包括：

[0097] (1)在接入应用的Jvm启动参数中配置，防刷限流Java Agent客户端，启动参数包括创建应用时获取到的应用id；

[0098] (2)防刷限流Java Agent客户端根据应用id到远程服务端获取当前应用所配置的

所有监控点防刷限流策略；

[0099] (3)服务端返回当前应用监控点的防刷限流策略；

[0100] (4)根据监控点信息，对加载的Java类进行逻辑入侵；

[0101] (5)启动监控点配置信息定时拉取任务；

[0102] (6)启动监控点统计信息定时上报任务，上报内容包括，周期时间的监控掉调用次数，防刷限流回调次数，异常调用测试等。

[0103] 图3c是本发明实施例三所提供的一种监控点调用流程示意图。如图3c所示，基于Java Agent的Java接口防刷限流中监控点的一次调用流程包括：

[0104] (1)根据Java类名称和方法名称，获取当前监控点配置的防刷限流策略

[0105] (2)获取防刷限流策略需要监控的入参属性名称

[0106] (3)根据监控点配置入参属性名称，获取入参中该属性名称所对应的值。

[0107] (4)判断该属性值是否在本地的防刷限流触发属性临时列表，如果在列表中，则根据防刷限流策略表所配置的回调结果进行降级熔断，如果不在则执行原有逻辑。

[0108] (5)异步上传当前属性值到服务端，以便服务端进行统计分析，并接受服务端的分析结果。

[0109] (6)客户端根据服务端返回的结果更新本地防刷限流触发属性临时列表。

[0110] 本发明实施例使用Java Agent技术在系统启动前完成监控点降级限流策略获取和监控点逻辑入侵，使得Java应用接入不需要进行代码改造；采用客户端步异上报监控属性值到服务端，再由服务端完成该监控属性值的统计分析，并通知客户端更新本地防刷限流触发属性临时列表，能够过尽可的减少入侵逻辑对原有接口的性能影响。

[0111] 实施例四

[0112] 图4是本发明实施例四所提供的一种接口访问控制装置的结构示意图。该接口访问控制装置可以采用软件和/或硬件的方式实现，例如该接口访问控制装置可以配置于客户端中。如图4所示，该装置包括语法信息获取模块410、监控入参获取模块420和访问请求控制模块430，其中：

[0113] 语法信息获取模块410，用于响应于检测到的访问请求，获取访问请求的语法信息；

[0114] 监控入参获取模块420，用于确定语法信息关联的监控配置入参属性，其中，监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的；

[0115] 访问请求控制模块430，用于获取监控配置入参属性关联的访问属性值，基于访问属性值和访问控制列表对访问请求进行控制。

[0116] 本发明实施例通过语法信息获取模块响应于检测到的访问请求，获取访问请求的语法信息；监控入参获取模块确定语法信息关联的监控配置入参属性，其中，监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的；访问请求控制模块获取监控配置入参属性关联的访问属性值，基于访问属性值和访问控制列表对访问请求进行控制，通过预先配置访问控制信息，在应用程序启动前从服务端获取并加载，使得接口的访问控制配置更加方便，减少了Java应用的接入复杂度，使得Java应用的大量接入得以实现。

[0117] 可选的，在上述方案的基础上，访问请求控制模块430具体用于：

[0118] 将访问属性值与访问控制列表中的控制属性值进行匹配；

[0119] 当访问属性值与控制属性值匹配成功时，根据预先配置的回调结果进行降级熔断，生成访问失败的响应信息进行展示；

[0120] 当访问属性值与控制属性值未匹配成功时，执行访问请求。

[0121] 可选的，在上述方案的基础上，装置还包括控制列表更新模块，用于：

[0122] 将访问属性值发送至服务端，以使服务端生成访问属性值关联的访问分析结果；

[0123] 接收服务端发送的访问分析结果，基于访问分析结果对访问控制列表进行更新。

[0124] 可选的，在上述方案的基础上，装置还包括配置信息加载模块，用于：

[0125] 应用程序启动前，Java Agent客户端根据客户端关联的应用标识从服务端中加载预先配置的访问控制配置信息；

[0126] 基于访问控制配置信息对加载的Java类进行逻辑入侵，完成Java类的加载。

[0127] 本发明实施例所提供的接口访问控制装置可执行本发明任意实施例所提供的接口访问控制方法，具备执行方法相应的功能模块和有益效果。

[0128] 实施例五

[0129] 图5是本发明实施例五所提供的一种接口访问控制装置的结构示意图。该接口访问控制装置可以采用软件和/或硬件的方式实现，例如该接口访问控制装置可以配置于服务端中。如图5所示，该装置包括信息获取请求模块510、目标配置信息模块520和响应信息生成模块530，其中：

[0130] 信息获取请求模块510，用于接收客户端发送的访问控制配置信息获取请求；

[0131] 目标配置信息模块520，用于根据访问控制配置信息获取请求对应的应用标识确定目标访问控制配置信息；

[0132] 响应信息生成模块530，用于基于目标访问控制配置信息生成获取请求响应信息发送至客户端。

[0133] 本发明实施例通过信息获取请求模块接收客户端发送的访问控制配置信息获取请求；目标配置信息模块根据访问控制配置信息获取请求对应的应用标识确定目标访问控制配置信息；响应信息生成模块基于目标访问控制配置信息生成获取请求响应信息发送至客户端，通过预先配置访问控制信息，在应用程序启动前向客户端下发访问控制信息，使得接口的访问控制配置更加方便，减少了Java应用的接入复杂度，使得Java应用的大量接入得以实现。

[0134] 可选的，在上述方案的基础上，装置还包括控制信息配置模块，用于：

[0135] 响应于检测到的访问控制信息配置请求，生成访问控制信息配置请求对应应用的访问控制信息配置界面并展示；

[0136] 响应于检测到的控制信息配置完成请求，获取控制信息配置完成请求关联的访问控制信息，并将访问控制信息与应用标识关联存储。

[0137] 本发明实施例所提供的接口访问控制装置可执行本发明任意实施例所提供的接口访问控制方法，具备执行方法相应的功能模块和有益效果。

[0138] 实施例六

[0139] 图6是本发明实施例六所提供的一种计算机设备的结构示意图。图6示出了适于用来实现本发明实施方式的示例性计算机设备612的框图。图6显示的计算机设备612仅仅是一个示例，不应对本发明实施例的功能和使用范围带来任何限制。

[0140] 如图6所示，计算机设备612以通用计算设备的形式表现。计算机设备612的组件可以包括但不限于：一个或者多个处理器616，系统存储器628，连接不同系统组件(包括系统存储器628和处理器616)的总线618。

[0141] 总线618表示几类总线结构中的一种或多种，包括存储器总线或者存储器控制器，外围总线，图形加速端口，处理器616或者使用多种总线结构中的任意总线结构的局域总线。举例来说，这些体系结构包括但不限于工业标准体系结构(ISA)总线，微通道体系结构(MAC)总线，增强型ISA总线、视频电子标准协会(VESA)局域总线以及外围组件互连(PCI)总线。

[0142] 计算机设备612典型地包括多种计算机系统可读介质。这些介质可以是任何能够被计算机设备612访问的可用介质，包括易失性和非易失性介质，可移动的和不可移动的介质。

[0143] 系统存储器628可以包括易失性存储器形式的计算机系统可读介质，例如随机存取存储器(RAM)630和/或高速缓存存储器632。计算机设备612可以进一步包括其它可移动/不可移动的、易失性/非易失性计算机系统存储介质。仅作为举例，存储装置634可以用于读写不可移动的、非易失性磁介质(图6未显示，通常称为“硬盘驱动器”)。尽管图6中未示出，可以提供用于对可移动非易失性磁盘(例如“软盘”)读写的磁盘驱动器，以及对可移动非易失性光盘(例如CD-ROM，DVD-ROM或者其它光介质)读写的光盘驱动器。在这些情况下，每个驱动器可以通过一个或者多个数据介质接口与总线618相连。存储器628可以包括至少一个程序产品，该程序产品具有一组(例如至少一个)程序模块，这些程序模块被配置以执行本发明各实施例的功能。

[0144] 具有一组(至少一个)程序模块642的程序/实用工具640，可以存储在例如存储器628中，这样的程序模块642包括但不限于操作系统、一个或者多个应用程序、其它程序模块以及程序数据，这些示例中的每一个或某种组合中可能包括网络环境的实现。程序模块642通常执行本发明所描述的实施例中的功能和/或方法。

[0145] 计算机设备612也可以与一个或多个外部设备614(例如键盘、指向设备、显示器624等)通信，还可与一个或者多个使得用户能与该计算机设备612交互的设备通信，和/或与使得该计算机设备612能与一个或多个其它计算设备进行通信的任何设备(例如网卡，调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口622进行。并且，计算机设备612还可以通过网络适配器620与一个或者多个网络(例如局域网(LAN)，广域网(WAN)和/或公共网络，例如因特网)通信。如图所示，网络适配器620通过总线618与计算机设备612的其它模块通信。应当明白，尽管图中未示出，可以结合计算机设备612使用其它硬件和/或软件模块，包括但不限于：微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储系统等。

[0146] 处理器616通过运行存储在系统存储器628中的程序，从而执行各种功能应用以及数据处理，例如实现本发明实施例所提供的接口访问控制方法，该方法包括：

[0147] 响应于检测到的访问请求，获取访问请求的语法信息；

[0148] 确定语法信息关联的监控配置入参属性，其中，监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的；

[0149] 获取监控配置入参属性关联的访问属性值，基于访问属性值和访问控制列表对访

问请求进行控制；

[0150] 和/或，实现本发明实施例所提供的接口访问控制方法，该方法包括：

[0151] 接收客户端发送的访问控制配置信息获取请求；

[0152] 根据访问控制配置信息获取请求对应的应用标识确定目标访问控制配置信息；

[0153] 基于目标访问控制配置信息生成获取请求响应信息发送至客户端。

[0154] 当然，本领域技术人员可以理解，处理器还可以实现本发明任意实施例所提供的接口访问控制方法的技术方案。

[0155] 实施例七

[0156] 本发明实施例七还提供了一种计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时实现本发明实施例所提供的接口访问控制方法，该方法包括：

[0157] 响应于检测到的访问请求，获取访问请求的语法信息；

[0158] 确定语法信息关联的监控配置入参属性，其中，监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的；

[0159] 获取监控配置入参属性关联的访问属性值，基于访问属性值和访问控制列表对访问请求进行控制；

[0160] 和/或，实现本发明实施例所提供的接口访问控制方法，该方法包括：

[0161] 接收客户端发送的访问控制配置信息获取请求；

[0162] 根据访问控制配置信息获取请求对应的应用标识确定目标访问控制配置信息；

[0163] 基于目标访问控制配置信息生成获取请求响应信息发送至客户端。

[0164] 当然，本发明实施例所提供的一种计算机可读存储介质，其上存储的计算机程序不限于如上的方法操作，还可以执行本发明任意实施例所提供的接口访问控制方法的相关操作。

[0165] 本发明实施例的计算机存储介质，可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括：具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中，计算机可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0166] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质，该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0167] 计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括——但不限于无线、电线、光缆、RF等等，或者上述的任意合适的组合。

[0168] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机

程序代码，程序设计语言包括面向对象的程序设计语言—诸如Java、Smalltalk、C++，还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络——包括局域网(LAN)或广域网(WAN)—连接到用户计算机，或者，可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0169] 注意，上述仅为本发明的较佳实施例及所运用技术原理。本领域技术人员会理解，本发明不限于这里的特定实施例，对本领域技术人员来说能够进行各种明显的变化、重新调整和替代而不会脱离本发明的保护范围。因此，虽然通过以上实施例对本发明进行了较为详细的说明，但是本发明不仅仅限于以上实施例，在不脱离本发明构思的情况下，还可以包括更多其他等效实施例，而本发明的范围由所附的权利要求范围决定。

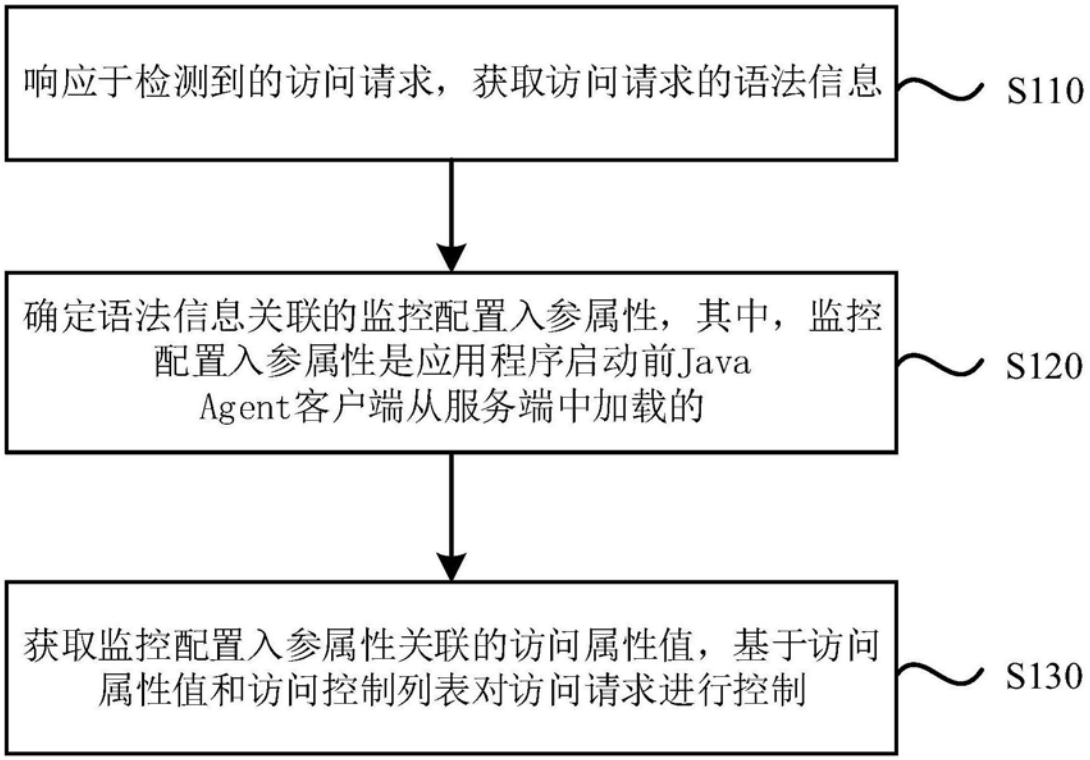

响应于检测到的访问请求，获取访问请求的语法信息
S110
确定语法信息关联的监控配置入参属性，其中，监控配置入参属性是应用程序启动前Java Agent客户端从服务端中加载的
S120
获取监控配置入参属性关联的访问属性值，基于访问属性值和访问控制列表对访问请求进行控制
S130

图1

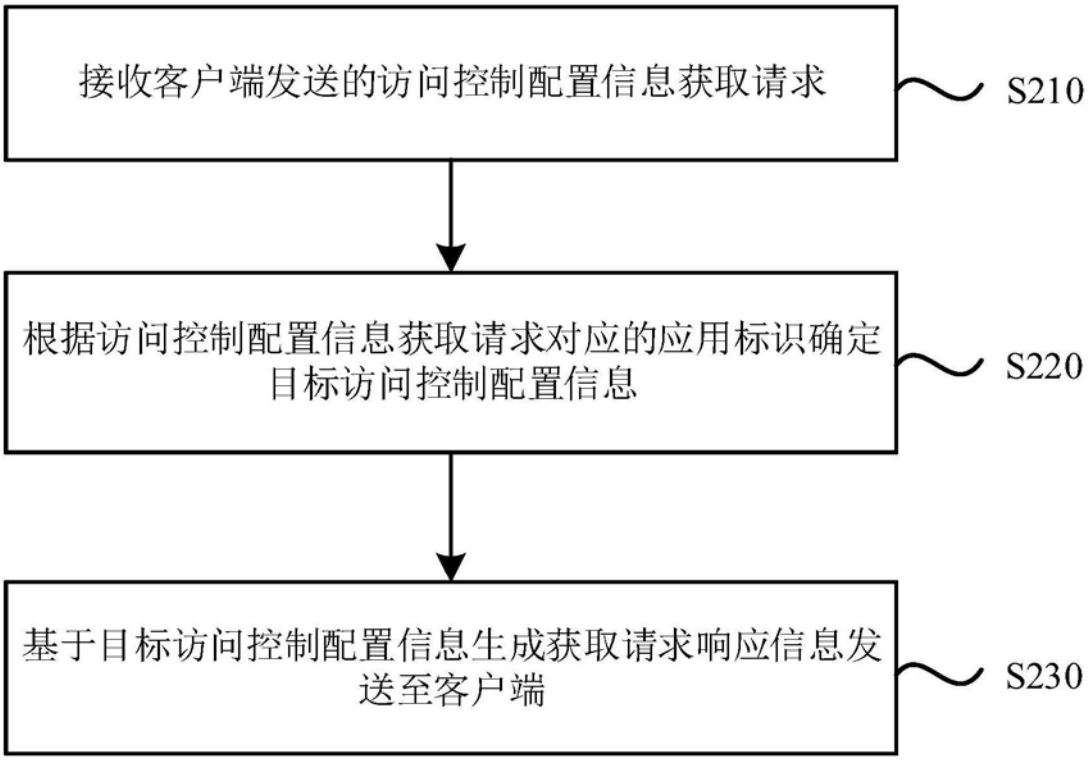

接收客户端发送的访问控制配置信息获取请求
S210
根据访问控制配置信息获取请求对应的应用标识确定目标访问控制配置信息
S220
基于目标访问控制配置信息生成获取请求响应信息发送至客户端
S230

图2

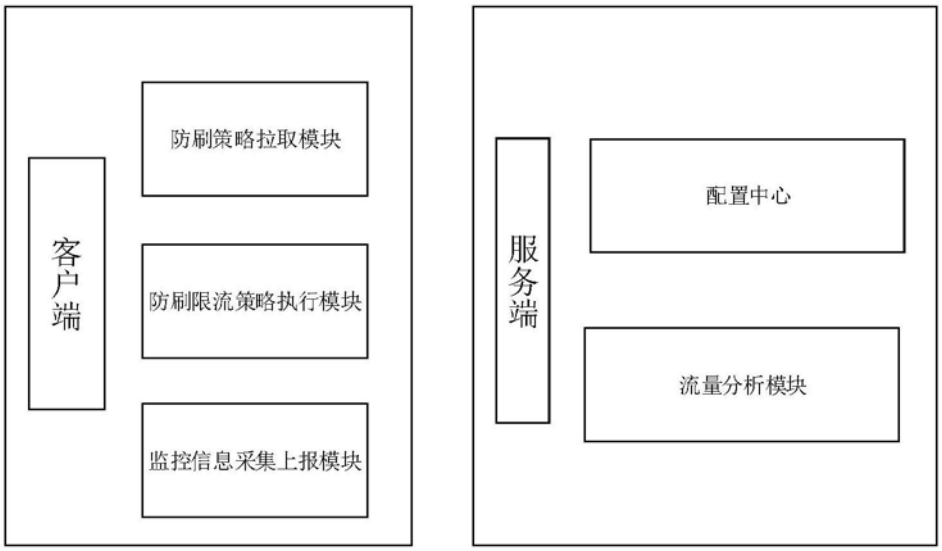


图3a

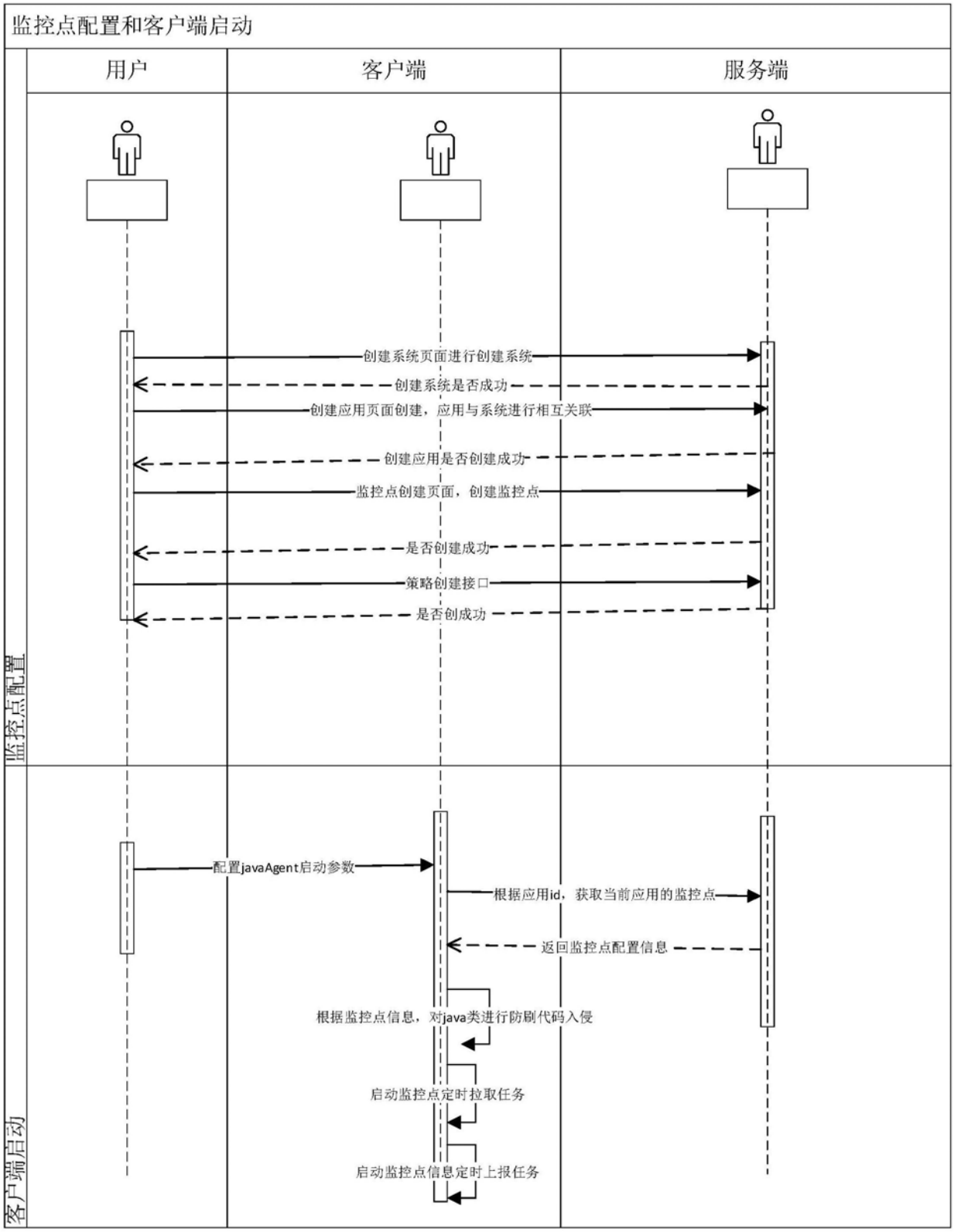

监控点配置和客户端启动
用户
客户端
服务端
监控点配置
创建系统页面进行创建系统
创建系统是否成功
创建应用页面创建，应用与系统进行相互关联
创建应用是否创建成功
监控点创建页面，创建监控点
是否创建成功
策略创建接口
是否创成功
客户端启动
配置javaAgent启动参数
根据应用id，获取当前应用的监控点
返回监控点配置信息
根据监控点信息，对java类进行防刷代码入侵
启动监控点定时拉取任务
启动监控点信息定时上报任务

图3b

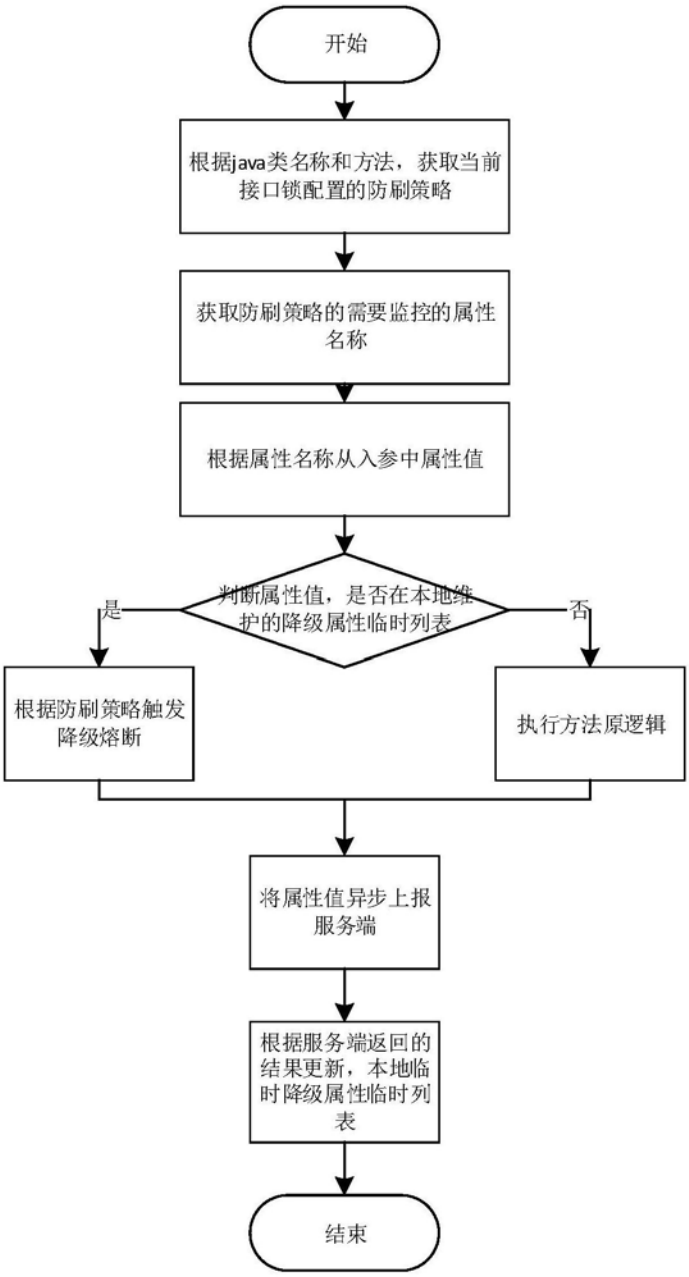

开始
根据java类名称和方法，获取当前接口锁配置的防刷策略
获取防刷策略的需要监控的属性名称
根据属性名称从入参中属性值
判断属性值，是否在本地维护的降级属性临时列表
是
否
根据防刷策略触发降级熔断
执行方法原逻辑
将属性值异步上报服务端
根据服务端返回的结果更新，本地临时降级属性临时列表
结束

图3c

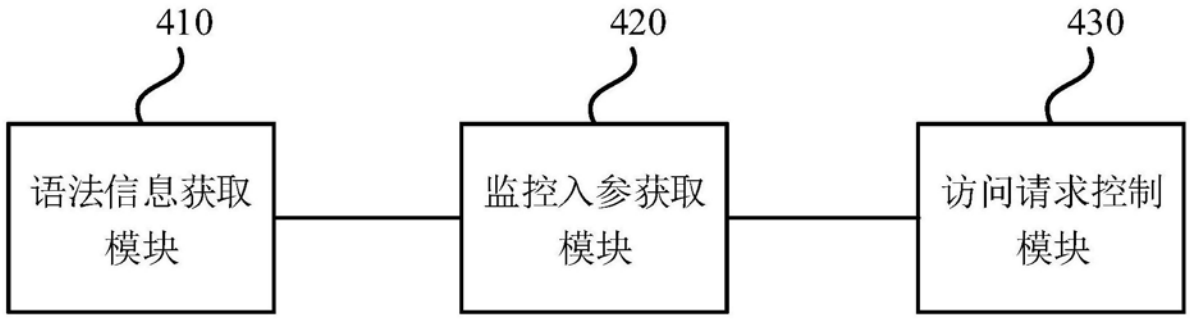

410
420
430
语法信息获取模块
监控入参获取模块
访问请求控制模块

图4

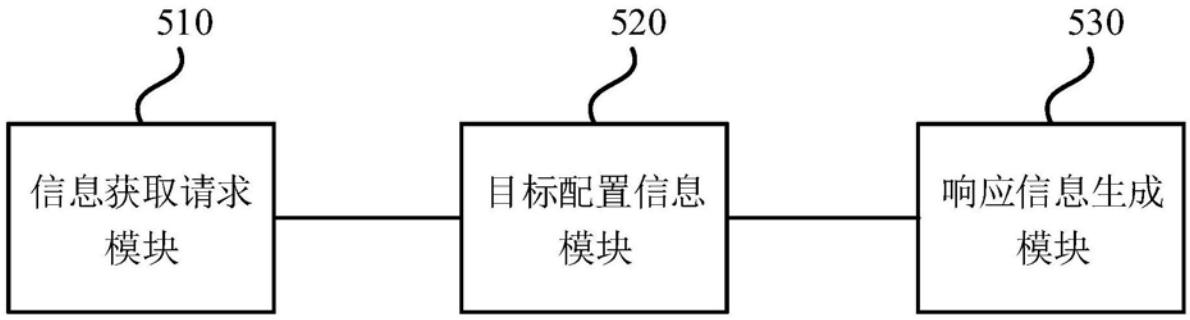

510
520
530
信息获取请求模块
目标配置信息模块
响应信息生成模块

图5

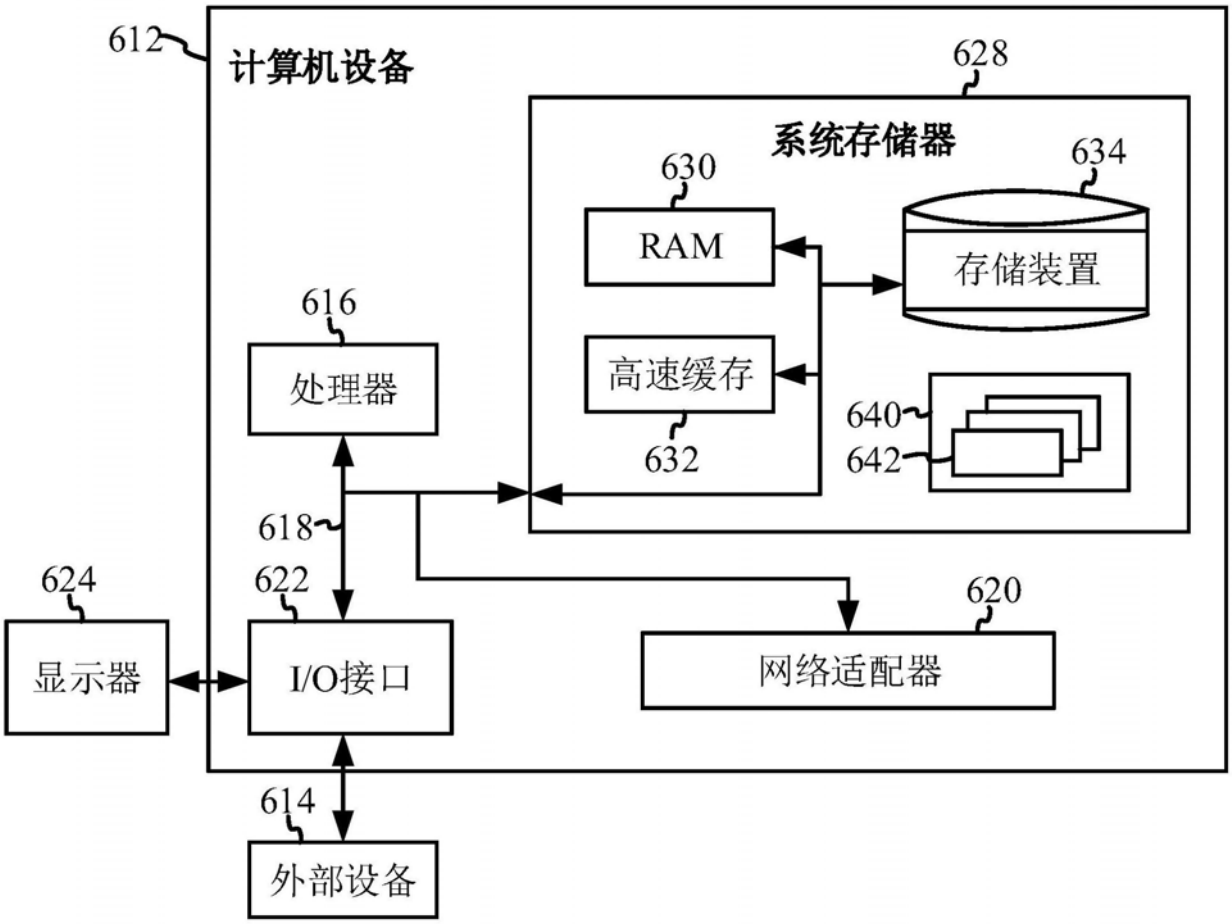

612
计算机设备
628
系统存储器
630
RAM
634
存储装置
616
处理器
高速缓存
632
640
642
618
624
622
620
显示器
I/O接口
网络适配器
614
外部设备

图6