



(12) 发明专利申请

(10) 申请公布号 CN 113938473 A

(43) 申请公布日 2022.01.14

(21) 申请号 202111187825.5

(22) 申请日 2021.10.12

(71) 申请人 平安银行股份有限公司

地址 518000 广东省深圳市罗湖区深南东路5047号

(72) 发明人 罗维练

(74) 专利代理机构 深圳国新南方知识产权代理有限公司 44374

代理人 李小东

(51) Int. Cl.

H04L 67/02 (2022.01)

H04L 41/045 (2022.01)

G06F 16/21 (2019.01)

G06F 16/25 (2019.01)

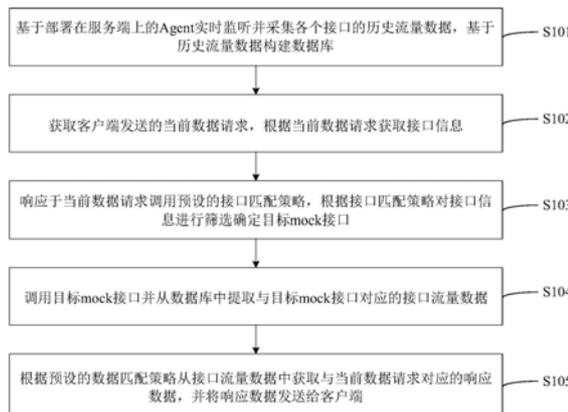
权利要求书2页 说明书11页 附图5页

(54) 发明名称

基于流量的自动Mock方法、装置、设备及存储介质

(57) 摘要

本发明涉及人工智能及数字医疗领域,具体公开了一种基于流量的自动Mock方法、装置、设备及存储介质。该方法包括:基于部署在服务端上的Agent实时监听并采集各个接口的历史流量数据,基于历史流量数据构建数据库;获取客户端发送的当前数据请求,根据当前数据请求获取接口信息;响应于当前数据请求调用预设的接口匹配策略,根据接口匹配策略对接口信息进行筛选确定目标mock接口;调用目标mock接口并从数据库中提取与目标mock接口对应的接口流量数据;根据预设的数据匹配策略从接口流量数据中获取与当前数据请求对应的响应数据,并将响应数据发送给客户端。本发明能够减少维护工作量以及提高数据数量效率。



1. 一种基于流量的自动Mock方法,其特征在于,包括:

基于部署在服务端上的Agent实时监听并采集各个接口的历史流量数据,基于所述历史流量数据构建数据库;

获取客户端发送的当前数据请求,根据所述当前数据请求获取接口信息;

响应于所述当前数据请求调用预设的接口匹配策略,根据所述接口匹配策略对所述接口信息进行筛选确定目标mock接口;

调用所述目标mock接口并从所述数据库中提取与所述目标mock接口对应的接口流量数据;

根据预设的数据匹配策略从所述接口流量数据中获取与所述当前数据请求对应的响应数据,并将所述响应数据发送给所述客户端。

2. 根据权利要求1所述的自动Mock方法,其特征在于,所述基于所述历史流量数据构建数据库包括:

对所述历史流量数据进行脱敏处理和数据清洗处理;

基于处理后的所述历史流量数据构建数据库并将处理后的所述历史流量数据存储于所述数据库中。

3. 根据权利要求1所述的自动Mock方法,其特征在于,所述响应于所述当前数据请求调用预设的接口匹配策略,根据所述接口匹配策略对所述接口信息进行筛选确定目标mock接口还包括:

响应于所述当前数据请求调用预设的接口匹配策略,所述接口匹配策略包括url匹配策略和参数匹配策略;

若所述接口匹配策略为所述url匹配策略,则根据所述接口信息获取url信息,根据所述url信息和所述url匹配策略确定目标mock接口;

若所述接口匹配策略为所述参数匹配策略,则根据所述接口信息获取参数信息,根据所述参数信息和所述参数匹配策略确定目标mock接口。

4. 根据权利要求1所述的自动Mock方法,其特征在于,所述响应于所述当前数据请求调用预设的接口匹配策略,根据所述接口匹配策略对所述接口信息进行筛选确定目标mock接口的步骤之后,还包括:

查询所述目标mock接口的开关状态;

若所述目标mock接口的开关状态为开启状态,则调用所述目标mock接口;

若所述目标mock接口的开关状态为关闭状态,则将所述当前数据请求转发至第三方平台以获取与所述当前数据请求对应的实时响应数据。

5. 根据权利要求1所述的自动Mock方法,其特征在于,所述根据预设的数据匹配策略从所述接口流量数据中获取与所述当前数据请求对应的响应数据,并将所述响应数据发送给所述客户端还包括:

调用预设的所述数据匹配策略,所述数据匹配策略包括参数最优策略、参数最多策略、最常用策略和最近使用策略;

将所述接口流量数据按照所述参数最优策略、所述参数最多策略、所述最常用策略以及所述最近使用策略的顺序与所述当前数据请求进行匹配计算,当最先计算得到匹配计算结果时,则不进行后续数据匹配策略,并将计算得到的匹配计算结果作为所述接口流量数

据的匹配计算结果；

根据所述匹配计算结果从所述接口流量数据中获得与所述当前数据请求相匹配的目标流量数据；

从所述目标流量数据中获取与所述目标数据请求对应的响应数据，并将所述响应数据发送给所述客户端。

6. 根据权利要求5所述的自动Mock方法，其特征在于，所述将所述接口流量数据按照所述参数最优策略、所述参数最多策略、所述最常用策略以及所述最近使用策略的顺序与所述当前数据请求进行匹配计算的步骤之后，还包括：

判断匹配计算结果是否存在；

若存在所述匹配计算结果，则当最先计算得到匹配计算结果时，则不进行后续数据匹配策略，并将计算得到的匹配计算结果作为所述接口流量数据的匹配计算结果；

若不存在所述匹配计算结果，则获取用户配置匹配策略，根据所述用户配置匹配策略确定目标流量数据。

7. 根据权利要求6所述的自动Mock方法，其特征在于，所述用户配置匹配策略包括所述参数最优策略、所述参数最多策略、所述最常用策略以及所述最近使用策略中的一种或多种。

8. 一种基于流量的自动Mock装置，其特征在于，包括：

采集模块，用于基于部署在服务端上的Agent实时监听并采集各个接口的历史流量数据，基于所述历史流量数据构建数据库；

获取模块，用于获取客户端发送的当前数据请求，根据所述当前数据请求获取接口信息；

第一匹配模块，用于响应于所述当前数据请求调用预设的接口匹配策略，根据所述接口匹配策略对所述接口信息进行筛选确定目标mock接口；

提取模块，用于调用所述目标mock接口并从所述数据库中提取与所述目标mock接口对应的接口流量数据；

第二匹配模块，用于根据预设的数据匹配策略从所述接口流量数据中获取与所述当前数据请求对应的响应数据，并将所述响应数据发送给所述客户端。

9. 一种计算机设备，包括：存储器、处理器及存储在存储器上并可在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1-7任一项所述的基于流量的自动Mock方法。

10. 一种计算机存储介质，其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1-7中任一项所述的基于流量的自动Mock方法。

基于流量的自动Mock方法、装置、设备及存储介质

技术领域

[0001] 本发明涉及人工智能技术领域,特别是涉及一种基于流量的自动Mock方法、装置、设备及存储介质。

背景技术

[0002] Mock服务根据用户配置匹配规则判断当前访问是否需要Mock,需要则在Mock平台上匹配配置的响应报文,如果匹配成功,则直接返回原在Mock平台上配置的匹配成功的响应报文,而不用访问第三方接口,如果匹配不成功,则由Mock服务转发到第三方接口,返回真实访问的结果。目前Mock方案中,用户必须清楚接口的请求参数、响应报文的格式,才可以配置符合要求的匹配规则,而对于不同场景的业务测试,需要配置不同的Mock匹配规则,如果接口繁多,那么配置工作量较大,后续业务调整后,也需要花费大量时间去维护,时间成本高,效率低,而且对于不同的接口,用户必须要清楚对应接口的请求报文和响应报文,否则无法配置参数匹配规则和需要的响应报文数据。

发明内容

[0003] 本发明提供一种基于流量的自动Mock方法、装置、设备及存储介质,能够解决规则配置的困难和业务调整而引起的大量维护工作,导致数据处理效率低的问题。

[0004] 为解决上述技术问题,本发明采用的一个技术方案是:提供一种基于流量的自动Mock方法,包括:

[0005] 基于部署在服务端上的Agent实时监听并采集各个接口的历史流量数据,基于所述历史流量数据构建数据库;

[0006] 获取客户端发送的当前数据请求,根据所述当前数据请求获取接口信息;

[0007] 响应于所述当前数据请求调用预设的接口匹配策略,根据所述接口匹配策略对所述接口信息进行筛选确定目标mock接口;

[0008] 调用所述目标mock接口并从所述数据库中提取与所述目标mock接口对应的接口流量数据;

[0009] 根据预设的数据匹配策略从所述接口流量数据中获取与所述当前数据请求对应的响应数据,并将所述响应数据发送给所述客户端。

[0010] 根据本发明的一个实施例,所述基于所述历史流量数据构建数据库还包括:

[0011] 对所述历史流量数据进行脱敏处理和数据清洗处理;

[0012] 基于处理后的所述历史流量数据构建数据库并将处理后的所述历史流量数据存储于所述数据库中。

[0013] 根据本发明的一个实施例,所述响应于所述当前数据请求调用预设的接口匹配策略,根据所述接口匹配策略对所述接口信息进行筛选确定目标mock接口还包括:

[0014] 响应于所述当前数据请求调用预设的接口匹配策略,所述接口匹配策略包括url匹配策略和参数匹配策略;

[0015] 若所述接口匹配策略为所述url匹配策略,则根据所述接口信息获取url信息,根据所述url信息和所述url匹配策略确定目标mock接口;

[0016] 若所述接口匹配策略为所述参数匹配策略,则根据所述接口信息获取参数信息,根据所述参数信息和所述参数匹配策略确定目标mock接口。

[0017] 根据本发明的一个实施例,所述响应于所述当前数据请求调用预设的接口匹配策略,根据所述接口匹配策略对所述接口信息进行筛选确定目标mock接口的步骤之后,还包括:

[0018] 查询所述目标mock接口的开关状态;

[0019] 若所述目标mock接口的开关状态为开启状态,则调用所述目标mock接口;

[0020] 若所述目标mock接口的开关状态为关闭状态,则将所述当前数据请求转发至第三方平台以获取与所述当前数据请求对应的实时响应数据。

[0021] 根据本发明的一个实施例,所述根据预设的数据匹配策略从所述接口流量数据中获取与所述当前数据请求对应的响应数据,并将所述响应数据发送给所述客户端还包括:

[0022] 调用预设的所述数据匹配策略,所述数据匹配策略包括参数最优策略、参数最多策略、最常用策略和最近使用策略;

[0023] 将所述接口流量数据按照所述参数最优策略、所述参数最多策略、所述最常用策略以及所述最近使用策略的顺序与所述当前数据请求进行匹配计算,当最先计算得到匹配计算结果时,则不进行后续数据匹配策略,并将计算得到的匹配计算结果作为所述接口流量数据的匹配计算结果;

[0024] 根据所述匹配计算结果从所述接口流量数据中获得与所述当前数据请求相匹配的目标流量数据;

[0025] 从所述目标流量数据中获取与所述目标数据请求对应的响应数据,并将所述响应数据发送给所述客户端。

[0026] 根据本发明的一个实施例,所述将所述接口流量数据按照所述参数最优策略、所述参数最多策略、所述最常用策略以及所述最近使用策略的顺序与所述当前数据请求进行匹配计算的步骤之后还包括:

[0027] 判断匹配计算结果是否存在;

[0028] 若存在所述匹配计算结果,则当最先计算得到匹配计算结果时,则不进行后续数据匹配策略,并将计算得到的匹配计算结果作为所述接口流量数据的匹配计算结果;

[0029] 若不存在所述匹配计算结果,则获取用户配置匹配策略,根据所述用户配置匹配策略确定目标流量数据。

[0030] 根据本发明的一个实施例,所述用户配置匹配策略包括所述参数最优策略、所述参数最多策略、所述最常用策略以及所述最近使用策略中的一种或多种。

[0031] 为解决上述技术问题,本发明采用的另一个技术方案是:提供一种基于流量的自动Mock装置,包括:

[0032] 采集模块,用于基于部署在服务端上的Agent实时监听并采集各个接口的历史流量数据,基于所述历史流量数据构建数据库;

[0033] 获取模块,用于获取客户端发送的当前数据请求,根据所述当前数据请求获取接口信息;

[0034] 第一匹配模块,用于响应于所述当前数据请求调用预设的接口匹配策略,根据所述接口匹配策略对所述接口信息进行筛选确定目标mock接口;

[0035] 提取模块,用于调用所述目标mock接口并从所述数据库中提取与所述目标mock接口对应的接口流量数据;

[0036] 第二匹配模块,用于根据预设的数据匹配策略从所述接口流量数据中获取与所述当前数据请求对应的响应数据,并将所述响应数据发送给所述客户端。

[0037] 为解决上述技术问题,本发明采用的再一个技术方案是:提供一种计算机设备,包括:存储器、处理器及存储在存储器上并可在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所述的基于流量的自动Mock方法。

[0038] 为解决上述技术问题,本发明采用的再一个技术方案是: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上述基于流量的自动Mock方法。

[0039] 本发明的有益效果是:通过接口匹配策略确定目标Mock接口,调用目标Mock接口并从数据库中提取对应的接口流量数据,能够满足不同参数或场景的自动Mock,减少了开发人员的工作量,降低了维护成本,提高维护效率,通过数据匹配策略从接口流量数据中获取与当前数据请求对应的响应数据实现自动Mock,提高数据处理效率,解决了规则配置的困难和业务调整而引起的大量维护工作,导致数据处理效率低的问题。

附图说明

[0040] 图1是本发明第一实施例的基于流量的自动Mock方法的流程示意图;

[0041] 图2是本发明实施例的基于流量的自动Mock方法中步骤S101的流程示意图;

[0042] 图3是本发明实施例的基于流量的自动Mock方法中步骤S103的流程示意图;

[0043] 图4是本发明实施例的基于流量的自动Mock方法中步骤S105的流程示意图;

[0044] 图5是本发明第二实施例的基于流量的自动Mock方法的流程示意图;

[0045] 图6是本发明实施例的基于流量的自动Mock装置的结构示意图;

[0046] 图7是本发明实施例的计算机设备的结构示意图;

[0047] 图8是本发明实施例的计算机存储介质的结构示意图。

具体实施方式

[0048] 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,所描述的实施例仅是本发明的一部分实施例,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。

[0049] 本发明中的术语“第一”、“第二”、“第三”仅用于描述目的,而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此,限定有“第一”、“第二”、“第三”的特征可以明示或者隐含地包括至少一个该特征。本发明的描述中,“多个”的含义是至少两个,例如两个,三个等,除非另有明确具体的限定。本发明实施例中所有方向性指示(诸如上、下、左、右、前、后……)仅用于解释在某一特定姿态(如附图所示)下各部件之间的相对位置关系、运动情况等,如果该特定姿态发生改变时,则该方向性指示也相应地随之改

变。此外，术语“包括”和“具有”以及它们任何变形，意图在于覆盖不排他的包含。例如包含了一系列步骤或单元的过程、方法、系统、产品或设备没有限定于已列出的步骤或单元，而是可选地还包括没有列出的步骤或单元，或可选地还包括对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0050] 在本文中提及“实施例”意味着，结合实施例描述的特定特征、结构或特性可以包含在本发明的至少一个实施例中。在说明书中的各个位置出现该短语并不一定均是指相同的实施例，也不是与其它实施例互斥的独立的或备选的实施例。本领域技术人员显式地和隐式地理解的是，本文所描述的实施例可以与其它实施例相结合。

[0051] 本申请实施例可以基于人工智能技术对相关的数据进行获取和处理。其中，人工智能(Artificial Intelligence, AI)是利用数字计算机或者数字计算机控制的机器模拟、延伸和扩展人的智能，感知环境、获取知识并使用知识获得最佳结果的理论、方法、技术及应用系统。

[0052] 人工智能基础技术一般包括如传感器、专用人工智能芯片、云计算、分布式存储、大数据处理技术、操作/交互系统、机电一体化等技术。人工智能软件技术主要包括计算机视觉技术、机器人技术、生物识别技术、语音处理技术、自然语言处理技术以及机器学习/深度学习等几大方向。

[0053] 图1是本发明第一实施例的基于流量的自动Mock方法的流程示意图。需注意的是，若有实质上相同的结果，本发明的方法并不以图1所示的流程顺序为限。如图1所示，该方法包括步骤：

[0054] 步骤S101：基于部署在服务端上的Agent实时监听并采集各个接口的历史流量数据，基于历史流量数据构建数据库。

[0055] 在步骤S101中，启动服务端应用服务，同时启动对应的Agent服务，Agent用于实时监听并采集对应接口的历史流量数据，历史流量数据包括请求数据、响应数据以及接口信息。在后续步骤中，通过接口信息匹配能够获取对应接口的请求数据以及与请求数据对应的响应数据，通过请求数据匹配能够获得关联的响应数据。本实施例基于历史流量数据构建数据库，数据库中存储的历史流量数据为Mock平台提供Mock数据，Mock平台可以为能够为客户端提供Mock数据的终端或程序模块。

[0056] 进一步地，请参见图2，步骤S101还包括以下步骤：

[0057] 步骤S201：基于部署在服务端上的Agent实时监听并采集各个接口的历史流量数据。

[0058] 步骤S202：对历史流量数据进行脱敏处理和数据清洗处理。

[0059] 在步骤S102中，从历史流量数据提取用户的敏感信息，敏感信息包括手机号码、姓名、身份证号码、卡号以及客户号等，将用户的敏感信息通过脱敏规则进行数据的变形，实现敏感隐私数据的可靠保护。进一步地，对脱敏处理后的历史流量数据进行数据清洗处理，数据清洗处理包括去重、筛选、过滤、降噪等操作。通过数据清洗处理进一步提高数据的可靠性，同时减少数据处理量，提高数据处理效率。

[0060] 步骤S203：基于处理后的历史流量数据构建数据库并将处理后的历史流量数据存储于数据库中。

[0061] 本实施例基于数据库，在获取相同或相似的数据请求时，可以从数据库中根据数

据请求直接调用对应的响应数据,从而进一步提高数据的复用率和数据处理效率。

[0062] 步骤S102:获取客户端发送的当前数据请求,根据当前数据请求获取接口信息。

[0063] 在步骤S102中,Mock平台可以包括多个不同的Mock接口,各个Mock接口均可设置Mock服务以及人工干预服务,各个不同的Mock接口适用相同的接口规则。Mock接口通过接口信息配置,接口信息包括url信息和参数信息。例如,当前数据请求的URL为http://xxx.xxx.xx/test/api?name=test&context=test,则接口的url信息为http://xxx.xxx.xx/test/api,参数信息为name=test&context=test。

[0064] 步骤S103:响应于当前数据请求调用预设的接口匹配策略,根据接口匹配策略对接口信息进行筛选确定目标mock接口。

[0065] 在步骤S103中,接口匹配策略包括url匹配策略和参数匹配策略,本实施例可以通过url匹配策略或参数匹配策略确定目标mock接口。url匹配策略即将接口信息中的url与所有Mock接口进行匹配,当接口信息中的url与Mock接口的url匹配成功,则将匹配成功的Mock接口确定为目标mock接口。参数匹配策略即将接口信息中的参数与所有Mock接口的参数进行匹配,当接口信息中的参数与Mock接口的参数匹配成功,则将匹配成功的Mock接口确定为目标mock接口。进一步地,本实施例确定目标mock接口之后,需查询目标mock接口的开关状态;若目标mock接口的开关状态为开启状态,则执行步骤S104;若目标mock接口的开关状态为关闭状态,则将当前数据请求转发至第三方平台以获取与当前数据请求对应的实时响应数据。

[0066] 进一步地,请参见图3,步骤S103还包括以下步骤:

[0067] 步骤S301:响应于当前数据请求调用预设的接口匹配策略,接口匹配策略包括url匹配策略和参数匹配策略。

[0068] 在步骤S301中,url匹配策略为将当前数据请求的url与接口url进行相似性匹配,匹配成功则将对应的接口确定为目标mock接口;参数匹配策略为将当前数据请求的参数与接口参数进行相似性匹配,匹配成功则将对应的接口确定为目标mock接口。

[0069] 步骤S302:若接口匹配策略为url匹配策略,则根据接口信息获取url信息,根据url信息和url匹配策略确定目标mock接口。

[0070] 在步骤S302中,为将当前数据请求的url与接口url进行相似性匹配,若url匹配成功,表明存在mock接口,则确定目标mock接口,若url匹配不成功,表明不存在mock接口,则将当前数据请求转发至第三方平台以获取与当前数据请求对应的实时响应数据。

[0071] 具体地,url匹配策略包括精确匹配或模糊匹配,例如,接口url为:http://xxx.xxx.xx/test/api,精确匹配时,url必须等于“/test/api”才能匹配成功,模糊匹配时,url=/test或url=/test/a或url=/tes均能匹配成功。

[0072] 步骤S303:若接口匹配策略为参数匹配策略,则根据接口信息获取参数信息,根据参数信息和参数匹配策略确定目标mock接口。

[0073] 在步骤S303中,将当前数据请求的参数与接口参数进行相似性匹配,若参数匹配成功,表明存在mock接口,则确定目标mock接口,若参数匹配不成功,表明不存在mock接口,则将当前数据请求转发至第三方平台以获取与当前数据请求对应的实时响应数据。

[0074] 本实施例的参数匹配包括url参数匹配和参数body匹配,每种匹配策略包括精确匹配或模糊匹配。具体地,例如,url参数匹配,url为:http://xxx.xxx.xx/test/api?name

=test&context=test,精确匹配需要配置name=test,或context=test,或name=test并且context=test,模糊匹配则可以配置name=tes或context=t或name=tes并且context=st等类似配置。精确匹配则要求配置里的每个值都必须完全等于,模糊匹配则要求配置里的每个值都必须包含即可。

[0075] 步骤S104:调用目标mock接口并从数据库中提取与目标mock接口对应的接口流量数据。

[0076] 在步骤S104中,以POST接口:http://xxx.xxx.xx/test/api作为实例,与该接口对应的接口流量数据有6条,接口流量数据列表具体如下:

id	ip	port	url	type	request	response	time	active	use_num
1	xxx	80	/test/api	POST	{ "id" : 10, "name" : "test", "context" : "good" }	{ "code" : 200, "message" : "test1" }	2021-08-23 19:20:30	1	20
2	xxx	80	/test/api	POST	{ "id" : 10, "context" : "good" }	{ "code" : 200, "message" : "test2" }	2021-08-23 19:20:31	1	10
3	xxx	80	/test/api	POST	{ "id" : 10, "name" : "test" }	{ "code" : 200, "message" : "test3" }	2021-08-23 19:20:32	1	120
4	xxx	80	/test/api	POST	{ "id" : 10, }	{ "code" : 200, "message" : "test4" }	2021-08-23 19:20:33	1	40
5	xxx	80	/test/api	POST	{ "id" : 14, "name" : "test", "context" : "good" }	{ "code" : 200, "message" : "test5" }	2021-08-23 19:20:34	1	90
6	xxx	80	/test/api	POST	{ "id" : 15, "context" : "good" }	{ "code" : 200, "message" : "test6" }	2021-08-23 19:20:35	1	70

[0077] [0078] 步骤S105:根据预设的数据匹配策略从接口流量数据中获取与当前数据请求对应的响应数据,并将响应数据发送给客户端。

[0079] 在步骤S105中,本实施例利用数据匹配策略将接口流量数据中的请求数据与当前数据请求进行匹配,若接口流量数据中存在与当前数据请求对应的请求数据,则匹配成功,将与该请求数据对应的响应数据作为当前数据请求的响应数据,并将响应数据返回给客户端。本实施的数据匹配策略包括参数最优策略、参数最多策略、最常用策略、最近使用策略,按照参数最优匹配策略、参数最多匹配策略、最常用匹配策略、最近使用匹配策略依序逐项进行匹配计算,直到匹配到响应数据则从响应数据中返回采集时间最新的一条响应数据给客户端,若后续仍有未进行的匹配策略则不再进行匹配计算,本实施例的参数最优策略和

参数最多策略中的参数匹配即将当前数据请求与上述接口流量数据列表中的“request”值进行比较。最常用匹配策略则选择上述接口流量数据列表中的“use_num”值最多的一条，最近使用匹配策略则选择上述接口流量数据列表中的“time”值最新的一条。

[0080] 进一步地，请参见图4，步骤S105还包括以下步骤：

[0081] 步骤S401：调用预设的数据匹配策略，数据匹配策略包括参数最优策略、参数最多策略、最常用策略和最近使用策略。

[0082] 在步骤S401中，参数最优策略为参数完全相等，即每个参数的值都一一相等，例如，当前数据请求的参数body为：

```
{
  "id": 10,
[0083]  "name": "test",
  "context": "good"
}
```

[0084] 则与上述接口流量数据列表的匹配计算结果为id、name、context值都相等的流量数据，即id=1的流量数据。

[0085] 本实施例的参数最多策略为参数不完全相等时，参数相等最多流量数据。例如请求参数body为：

```
{
[0086]  "name": "test",
  "context" : "good",
[0087]  "active": 1
}
```

[0088] 则与上述接口流量数据列表的匹配计算结果为id为1和5的流量数据。

[0089] 本实施例的最常用策略为历史被使用的次数最多的那条记录。例如，上述接口流量数据列表中使用次数use_num最大的是120，对应记录id为3的流量数据。

[0090] 本实施例的最近使用策略为采集时间最新的记录，例如，上述接口流量数据列表中的id为6的流量数据。

[0091] 步骤S402：将接口流量数据按照参数最优策略、参数最多策略、最常用策略以及最近使用策略的顺序与当前数据请求进行匹配计算，当最先计算得到匹配计算结果时，则不进行后续数据匹配策略，并将计算得到的匹配计算结果作为接口流量数据的匹配计算结果。

[0092] 在步骤S402中，目标流量数据包括请求数据和响应数据，目标流量数据可以为每项策略的匹配计算结果进行汇总后的结果，即各个策略之间不存在关联关系，也可以为依次经过参数最优策略、参数最多策略、最常用策略以及最近使用策略层层递进筛选出来的结果，即在参数最优策略的匹配计算上执行参数最多策略，在参数最多策略的结果上执行最常用策略或最近使用策略。若匹配计算结果不止一个时，则将最新的采集时间对应的接口流量数据作为最后的匹配计算结果。但是，在本实施例中，在进行匹配时，只要在一个策

略中获得匹配计算结果,则不再继续进行后续策略的匹配计算,因此,若参数最优策略和/或参数最多策略中获得匹配计算结果,则不执行最常用策略和/或最近使用策略,在参数最优策略和/或参数最多策略中的匹配计算结果往往不止一个,若参数最优策略和/或参数最多策略中未获得匹配计算结果,则继续进行最常用策略和/或最近使用策略,最常用策略和最近使用策略的匹配计算结果通常仅有一个。

[0093] 步骤S403:根据匹配计算结果从接口流量数据中获得与当前数据请求相匹配的目标流量数据。

[0094] 在步骤S403中,如上表所示,将表中6条接口流量数据按照参数最优策略与当前数据请求进行匹配计算,即按照参数最优策略将当前数据请求与表中的“request”参数进行匹配,若当前数据请求的参数body为:

```
{
  "id": 10,
[0095]  "name": "test",
  "context": "good"
},
```

[0096] 则匹配计算结果为id=1的接口流量数据,即id=1的接口流量数据为与当前数据请求相匹配的目标流量数据。

[0097] 步骤S404:从目标流量数据中获取与目标数据请求对应的响应数据,并将响应数据发送给客户端。

[0098] 在步骤S404中,以id=1的接口流量数据为与当前数据请求相匹配的目标流量数据,则id=1的接口流量数据中“response”为对应的响应数据。本实施例若目标流量数据具有多个,则获取目标流量数据的采集时间,将最新的采集时间对应的目标流量数据中的响应数据发送给客户端。

[0099] 本发明第一实施例的基于流量的自动Mock方法通过接口匹配策略确定目标Mock接口,调用目标Mock接口并从数据库中提取对应的接口流量数据,能够满足不同参数或场景的自动Mock,减少了开发人员的工作量,降低了维护成本,提高维护效率,通过数据匹配策略从接口流量数据中获取与当前数据请求对应的响应数据实现自动Mock,提高数据处理效率,解决了规则配置的困难和业务调整而引起的大量维护工作,导致数据处理效率低的问题。

[0100] 图5是本发明第二实施例的基于流量的自动Mock方法的流程示意图。需注意的是,若有实质上相同的结果,本发明的方法并不以图5所示的流程顺序为限。如图5所示,该方法包括步骤:

[0101] 步骤S501:基于部署在服务端上的Agent实时监听并采集各个接口的历史流量数据,基于历史流量数据构建数据库。

[0102] 在本实施例中,图5中的步骤S501和图1中的步骤S101类似,为简约起见,在此不再赘述。

[0103] 步骤S502:获取客户端发送的当前数据请求,根据当前数据请求获取接口信息。

[0104] 在本实施例中,图5中的步骤S502和图1中的步骤S102类似,为简约起见,在此不再

赘述。

[0105] 步骤S503:响应于当前数据请求调用预设的接口匹配策略,根据接口匹配策略对接口信息进行筛选确定目标mock接口。

[0106] 在本实施例中,图5中的步骤S503和图1中的步骤S103类似,为简约起见,在此不再赘述。

[0107] 步骤S504:调用目标mock接口并从数据库中提取与目标mock接口对应的接口流量数据。

[0108] 在本实施例中,图5中的步骤S504和图1中的步骤S104类似,为简约起见,在此不再赘述。

[0109] 步骤S505:调用预设的数据匹配策略,数据匹配策略包括参数最优策略、参数最多策略、最常用策略和最近使用策略。

[0110] 在本实施例中,图5中的步骤S505和图4中的步骤S401类似,为简约起见,在此不再赘述。

[0111] 步骤S506:将接口流量数据按照参数最优策略、参数最多策略、最常用策略以及最近使用策略的顺序与当前数据请求进行匹配计算。

[0112] 在本实施例中,图5中的步骤S506和图4中的步骤S402类似,为简约起见,在此不再赘述。

[0113] 步骤S507:判断匹配计算结果是否存在。

[0114] 在步骤S507,在逐项匹配过程中,若产生匹配计算结果,则执行步骤S508,再执行步骤S509、S511;若未产生匹配计算结果,则执行步骤S510,再执行步骤S511。

[0115] 步骤S508:若存在匹配计算结果,根据当最先计算得到匹配计算结果时,则不进行后续数据匹配策略,并将计算得到的匹配计算结果作为接口流量数据的匹配计算结果。

[0116] 在本实施例中,图5中的步骤S508和图4中的步骤S402类似,为简约起见,在此不再赘述。

[0117] 步骤S509:根据匹配计算结果从接口流量数据中获得与当前数据请求相匹配的目标流量数据。

[0118] 在本实施例中,图5中的步骤S509和图4中的步骤S403类似,为简约起见,在此不再赘述。

[0119] 步骤S510:若不存在匹配计算结果,则获取用户配置匹配策略,根据用户配置匹配策略确定目标流量数据。

[0120] 在步骤S510中,若不存在匹配计算结果,则获取用户配置匹配策略,根据用户配置匹配策略从接口流量数据中获取与当前数据请求对应的目标数据请求,并将目标数据请求对应的接口流量数据确定为目标流量数据。本实施例的用户配置匹配策略为用户手动进行选择,用户配置匹配策略包括参数最优策略、参数最多策略、最常用策略以及最近使用策略的一个或多个,当用户配置匹配策略包括多个时,按照优先级从高到底即参数最优策略、参数最多策略、最常用策略、最近使用策略逐项进行匹配计算。匹配计算可能会存在多条流量数据,进一步地,在返回响应数据过程中,用户可以手动勾选预设的返回策略,利用返回策略进一步对匹配计算进行筛选,返回策略例如:参数匹配返回、自动返回,按规则返回,其中,参数匹配返回包括参数最优、参数最多,按规则返回包括最近使用和最常使用,自动返

回则默认返回采集时间最新的一条响应数据。

[0121] 进一步地,响应数据设置支持固定报文、函数动态报文、数据库动态报文三种设置,

[0122] 固定报文:即直接返回设置的固定报文数据。

[0123] 函数动态报文:根据系统提供的函数格式设置,mock自动解析函数格式内容,转换为函数计算,然后把计算得到的值替换报文数据返回。

[0124] 数据库动态报文:据配置的数据库sql,动态查询返回相应报文。

[0125] 进一步地,在获取用户配置匹配策略之前还包括获取人工干预状态,若人工干预开启,则获取用户配置匹配策略,若人工干预关闭,则转发当前数据请求至第三方平台以获取与当前数据请求对应的实时响应数据。

[0126] 进一步地,步骤S510之后还包括,将响应数据存储于数据库中,作为下次可自动mock的数据源。

[0127] 步骤S511:从目标流量数据中获取与目标数据请求对应的响应数据,并将响应数据发送给客户端。

[0128] 在本实施例中,图5中的步骤S511和图4中的步骤S404类似,为简约起见,在此不再赘述。

[0129] 本发明第二实施例的基于流量的自动Mock方法在第一实施例的基础上,通过在匹配计算结果不存在时,增加了用户配置匹配策略,提高了数据获取的灵活性和适用范围。

[0130] 图6是本发明实施例的基于流量的自动Mock装置的结构示意图。如图6所示,该装置60包括采集模块61、获取模块62、第一匹配模块63、提取模块64以及第二匹配模块65。

[0131] 采集模块61用于基于部署在服务端上的Agent实时监听并采集各个接口的历史流量数据,基于历史流量数据构建数据库;

[0132] 获取模块62用于获取客户端发送的当前数据请求,根据当前数据请求获取接口信息;

[0133] 第一匹配模块63用于响应于当前数据请求调用预设的接口匹配策略,根据接口匹配策略对接口信息进行筛选确定目标mock接口;

[0134] 提取模块64用于调用目标mock接口并从数据库中提取与目标mock接口对应的接口流量数据;

[0135] 第二匹配模块65用于根据预设的数据匹配策略从接口流量数据中获取与当前数据请求对应的响应数据,并将响应数据发送给客户端。

[0136] 请参阅图7,图7为本发明实施例的计算机设备的结构示意图。如图7所示,该计算机设备70包括处理器71及和处理器71耦接的存储器72。

[0137] 存储器72存储有用于实现上述任一实施例所述的基于流量的自动Mock方法的程序指令。

[0138] 处理器71用于执行存储器72存储的程序指令以实现自动Mock。

[0139] 其中,处理器71还可以称为CPU(Central Processing Unit,中央处理单元)。处理器71可能是一种集成电路芯片,具有信号的处理能力。处理器71还可以是通用处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现成可编程门阵列(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。通用处理器可以是微处理器或者该处理器

也可以是任何常规的处理器等。

[0140] 参阅图8,图8为本发明实施例的计算机存储介质的结构示意图。本发明实施例的计算机存储介质存储有能够实现上述所有方法的程序文件81,其中,该程序文件81可以以软件产品的形式存储在上述计算机存储介质中,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方式所述方法的全部或部分步骤。而前述的计算机存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质,或者是计算机、服务器、手机、平板等终端设备。

[0141] 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,单元的划分,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,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0142] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0143] 以上仅为本发明的实施方式,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本发明的专利保护范围内。

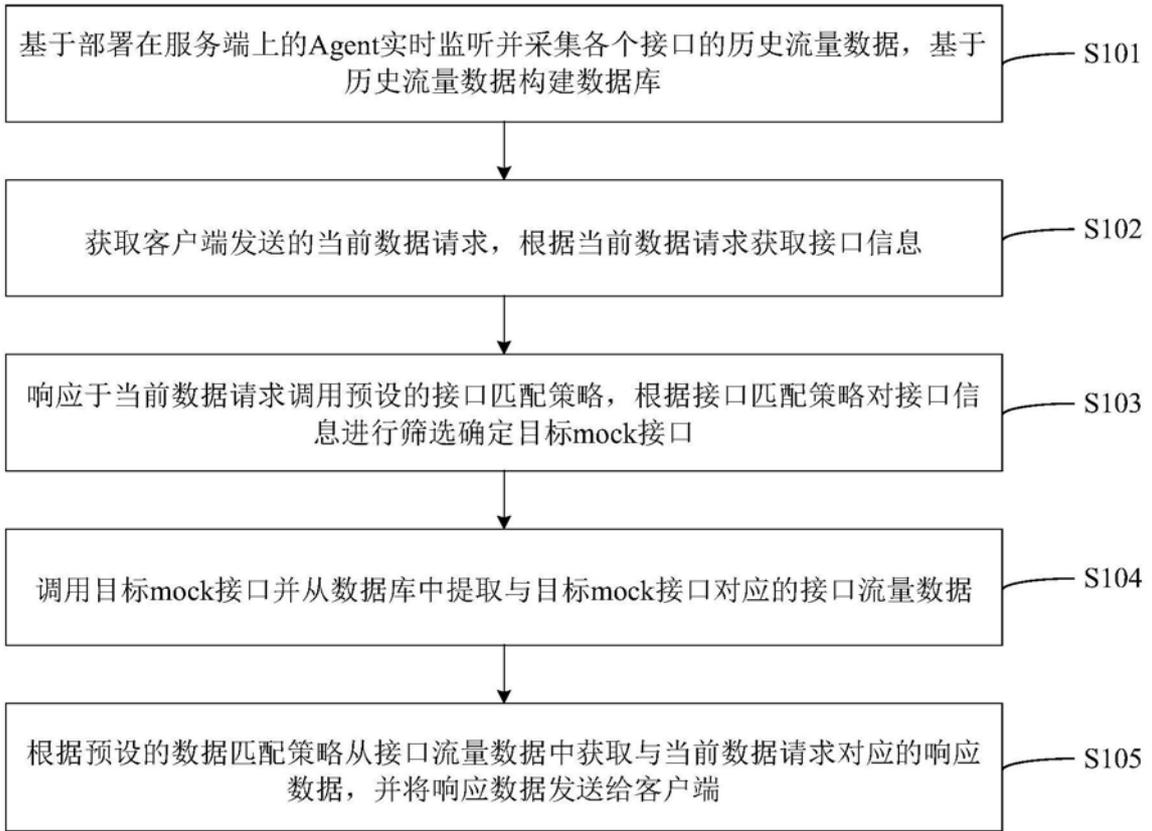


图1

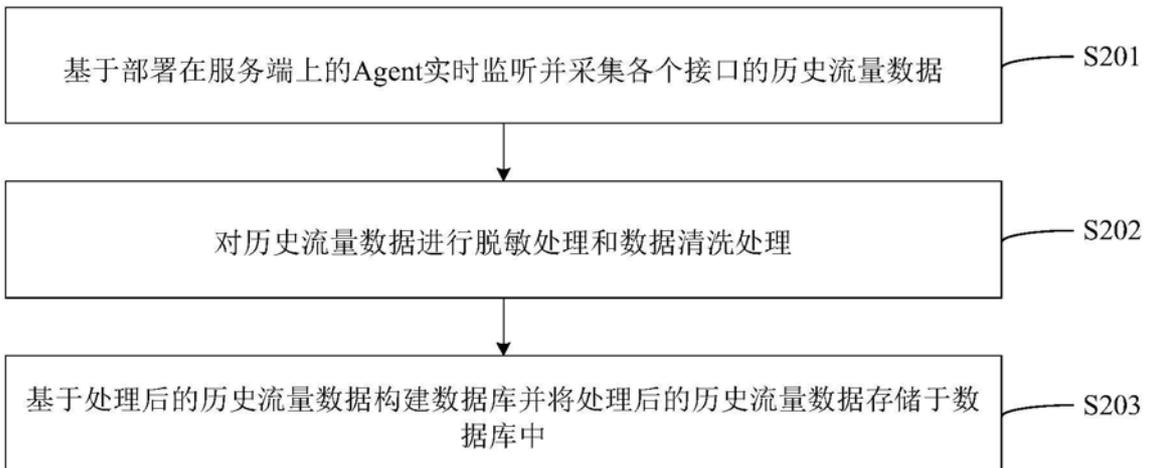


图2

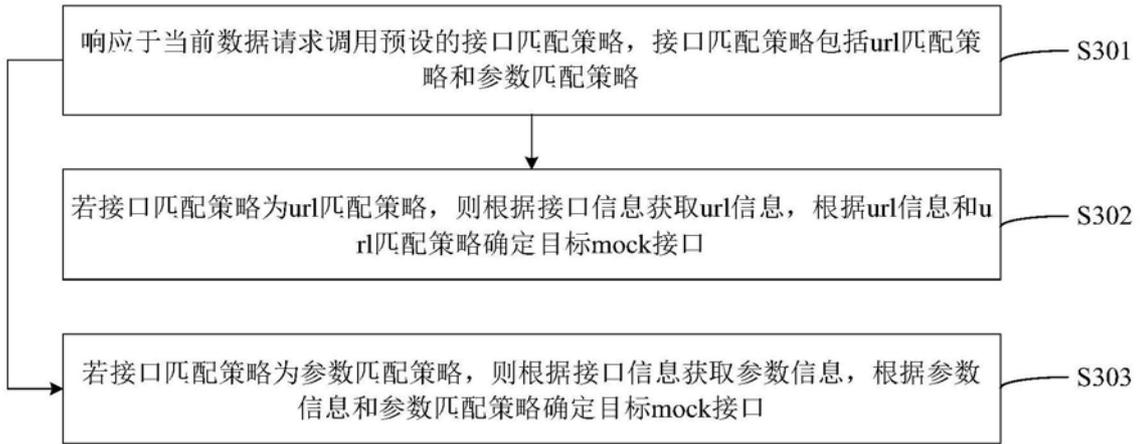


图3

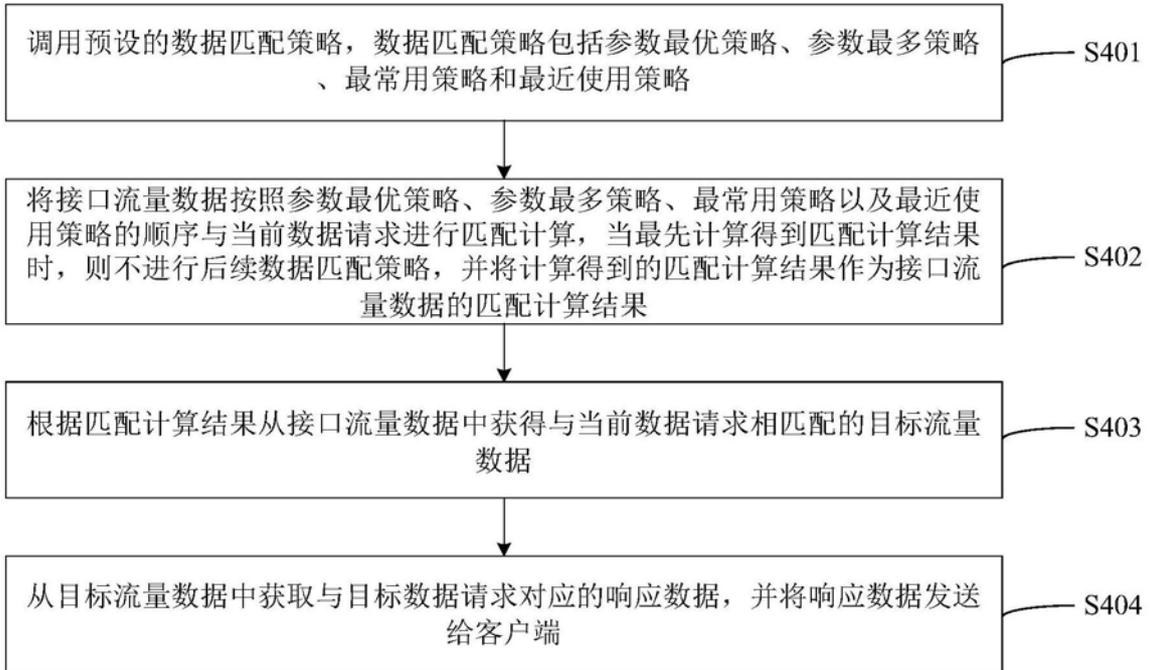


图4

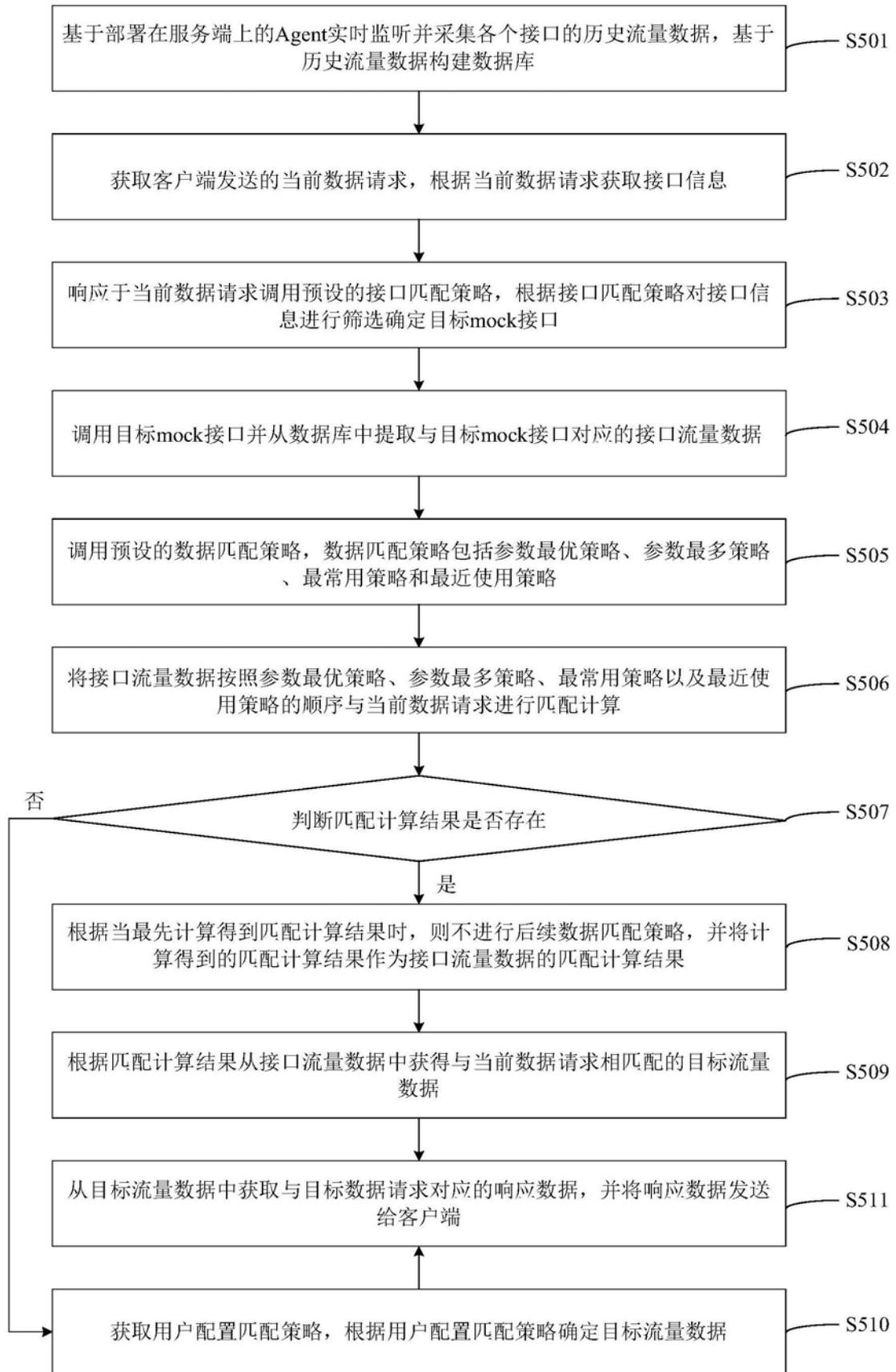


图5

60

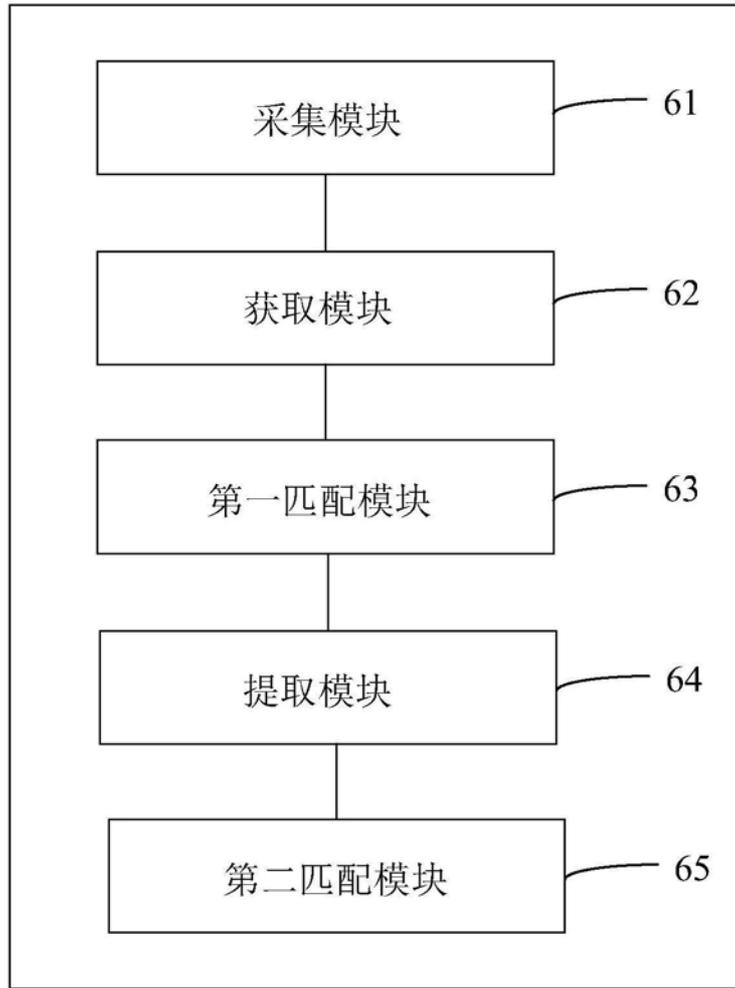


图6

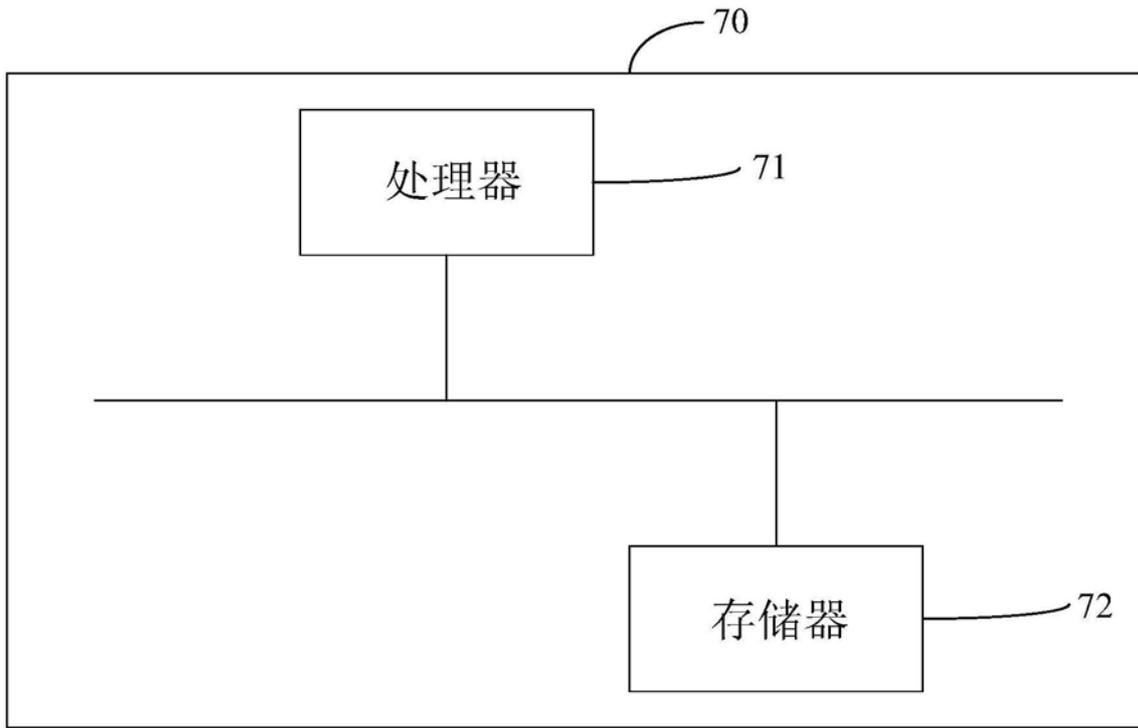


图7

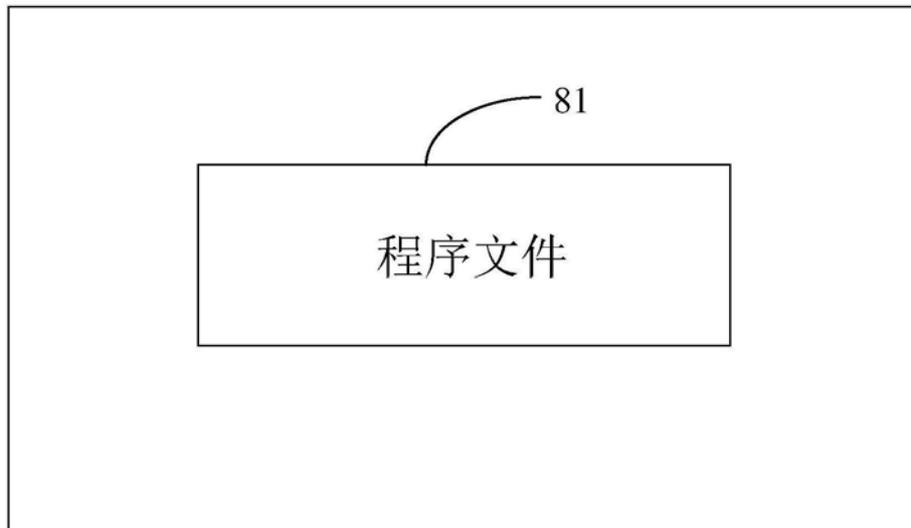


图8