



(12) 发明专利

(10) 授权公告号 CN 110851159 B

(45) 授权公告日 2023. 01. 10

(21) 申请号 201911050248.8

H04L 67/60 (2022.01)

(22) 申请日 2019.10.31

(56) 对比文件

(65) 同一申请的已公布的文献号  
申请公布号 CN 110851159 A

CN 109766124 A, 2019.05.17

CN 109787882 A, 2019.05.21

CN 109918079 A, 2019.06.21

(43) 申请公布日 2020.02.28

CN 107180338 A, 2017.09.19

(73) 专利权人 浙江大搜车软件技术有限公司  
地址 310000 浙江省杭州市余杭区五常街  
道五常大道165号2幢911室

US 2015365308 A1, 2015.12.17

US 2017322777 A1, 2017.11.09

审查员 罗思异

(72) 发明人 叶海宁

(74) 专利代理机构 杭州华进联浙知识产权代理  
有限公司 33250  
专利代理师 金无量

(51) Int. Cl.

G06F 8/65 (2018.01)

H04L 67/1097 (2022.01)

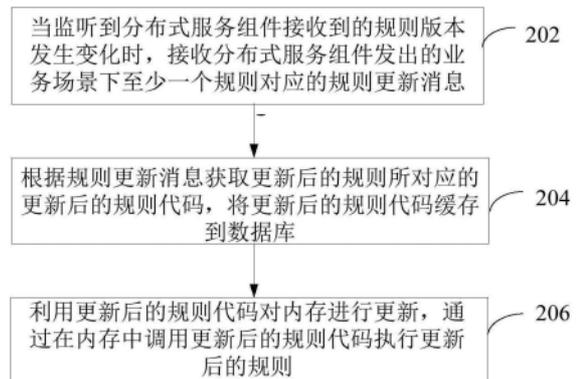
权利要求书2页 说明书8页 附图3页

(54) 发明名称

业务规则更新方法、装置、计算机设备和存储介质

(57) 摘要

本发明涉及一种业务规则更新方法、装置、计算机设备和存储介质,所述方法包括:当监听到分布式服务组件接收到的规则版本发生变化时,接收所述分布式服务组件发出的业务场景下至少一个规则对应的规则更新消息;根据所述规则更新消息获取更新后的规则所对应的更新后的规则代码,将所述更新后的规则代码缓存到数据库;利用所述更新后的规则代码对内存进行更新,通过在内存中调用所述更新后的规则代码执行更新后的规则。采用本方法使得业务系统能够快速自动更新业务规则代码,节省人工更新代码中不必要的操作环节。



1. 一种业务规则更新方法,所述方法包括:

当监听到分布式服务组件接收到的规则版本发生变化时,接收所述分布式服务组件发出的业务场景下至少一个规则对应的规则更新消息;

根据所述规则更新消息获取更新后的规则所对应的更新后的规则代码,包括:在所述规则更新消息中提取与所述更新后的规则对应的场景标识、版本标识;所述版本标识为当前使用版本;根据所述场景标识、所述当前使用版本获取所述更新后的规则代码;将所述更新后的规则代码缓存到数据库;

利用所述更新后的规则代码对内存进行更新,通过在内存中调用所述更新后的规则代码执行更新后的规则;一个所述业务场景支持多个相同逻辑的规则,并对多个规则进行优先级的排序;

当接收到业务请求时,根据所述业务请求执行更新后的规则代码所对应的业务场景;

通过业务系统调用更新后的规则代码执行所述业务请求;

根据所述业务方请求中包含的场景标识,规则标识匹配到终端提供的相应的业务场景调用接口;

当所述业务方调用并执行相应的业务场景时,所述业务系统通过规则引擎解析相应的规则代码,执行所述规则代码所对应的业务场景;

对所述业务场景所对应的规则调用次数进行统计。

2. 根据权利要求1所述的方法,其特征在于,所述分布式服务组件包括Zookeeper,所述方法还包括:

通过页面代码编辑器接收更新后的规则代码,存入数据库;

将所述更新后的规则代码对应的规则版本设置为当前使用版本;

将所述Zookeeper中将更新前的规则版本变更为所述当前使用版本。

3. 根据权利要求2所述的方法,其特征在于,所述通过页面代码编辑器接收更新后的规则代码后,所述方法还包括:

接收测试请求,对所述测试请求返回测试页面;

接收所述测试页面中的测试参数,执行所述测试请求;

当所述执行测试请求的返回值和预期结果一致,保存所述更新后的规则代码至数据库。

4. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

当所述更新后的规则代码在执行过程中出现异常时,获取更新前的规则代码对应的可用版本;

从当前使用版本回退至更新前的规则代码对应的可用版本。

5. 根据权利要求1所述的方法,其特征在于,所述方法还包括:所述规则代码为新增代码,所述终端通过新增创建一个所述业务场景,在业务场景下通过新增创建规则。

6. 一种业务规则更新装置,其特征在于,所述装置包括:

接收模块,用于当监听到分布式服务组件接收到的规则版本发生变化时,接收所述分布式服务组件发出的业务场景下至少一个规则对应的规则更新消息;

接收模块还用于当接收到业务请求时,根据所述业务请求执行更新后的规则代码所对应的业务场景;更新模块,用于根据所述规则更新消息获取更新后的规则所对应的更新后

的规则代码,包括:在所述规则更新消息中提取与所述更新后的规则对应的场景标识、版本标识;所述版本标识为当前使用版本;根据所述场景标识、所述当前使用版本获取所述更新后的规则代码;将所述更新后的规则代码缓存到数据库;

调用模块,用于利用所述更新后的规则代码对内存进行更新,通过在内存中调用所述更新后的规则代码执行更新后的规则;一个所述业务场景支持多个相同逻辑的规则,并对多个规则进行优先级的排序;

调用模块还用于通过业务系统调用更新后的规则代码执行所述业务请求;根据所述业务方请求中包含的场景标识,规则标识匹配到终端提供的相应的业务场景调用接口;

执行模块,用于当业务方调用并执行相应的业务场景时,所述业务系统通过规则引擎解析相应的规则代码,执行规则代码所对应的业务场景;

统计模块,用于对所述业务场景所对应的规则调用次数进行统计。

7. 根据权利要求6所述的装置,其特征在于,所述接收模块还用于通过页面代码编辑器接收更新后的规则代码,存入数据库;所述更新模块还用于将所述更新后的规则代码对应的规则版本设置为当前使用版本;将Zookeeper中更新前的规则版本变更为所述当前使用版本。

8. 根据权利要求7所述的装置,其特征在于,接收模块还用于接收测试请求,对所述测试请求返回测试页面;接收所述测试页面中的测试参数,执行所述测试请求;

更新模块,还用于当所述执行测试请求的返回值和预期结果一致,保存所述更新后的规则代码至数据库。

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权利要求1至5中任一项所述方法的步骤。

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,所述计算机程序被处理器执行时实现权利要求1至5中任一项所述的方法的步骤。

## 业务规则更新方法、装置、计算机设备和存储介质

### 技术领域

[0001] 本申请涉及计算机技术领域,特别是涉及一种业务规则更新方法、装置、计算机设备和存储介质。

### 背景技术

[0002] 随着计算机技术的发展,业务系统提供了各种业务功能。业务功能通过业务规则的代码可以实现。

[0003] 在传统方式中,业务系统经常需要变更业务规则。有时候只是一个小的业务规则的变化,也需要经过完整的人工操作的流程,给开发人员增加了不必要的工作,降低了业务规则的更新效率。因此,如何减不必要的人工工作,提高业务规则更新效率成为目前需要解决的一个技术问题。

### 发明内容

[0004] 基于此,有必要针对上述技术问题,提供一种能够减少不必要的人工工作,提高业务规则更新效率的业务规则更新方法、装置、计算机设备和存储介质。

[0005] 一种业务规则更新方法,所述方法包括:

[0006] 当监听到分布式服务组件接收到的规则版本发生变化时,接收所述分布式服务组件发出的业务场景下至少一个规则对应的规则更新消息;

[0007] 根据所述规则更新消息获取更新后的规则所对应的更新后的规则代码,将所述更新后的规则代码缓存到数据库;

[0008] 利用所述更新后的规则代码对内存进行更新,通过在内存中调用所述更新后的规则代码执行更新后的规则。

[0009] 在其中一个实施例中,所述分布式服务组件包括Zookeeper,所述方法还包括:

[0010] 通过页面代码编辑器接收更新后的规则代码,存入数据库;

[0011] 将所述更新后的规则代码对应的规则版本设置为当前使用版本;

[0012] 将所述Zookeeper中将更新前的规则版本变更为所述当前使用版本。

[0013] 在其中一个实施例中,所述通过页面代码编辑器接收更新后的规则代码后,所述方法还包括:

[0014] 接收测试请求,对所述测试请求返回测试页面;

[0015] 接收所述测试页面中的测试参数,执行所述测试请求;

[0016] 当所述执行测试请求的返回值和预期结果一致,保存所述更新后的规则代码至数据库。

[0017] 在其中一个实施例中,所述根据所述规则更新消息获取更新后的规则所对应的更新后的规则代码包括:

[0018] 在所述规则更新消息中提取与所述更新后的规则对应的场景标识、版本标识;所述版本标识为当前使用版本;

- [0019] 根据所述场景标识、所述当前使用版本获取所述更新后的规则代码。
- [0020] 在其中一个实施例中,所述方法还包括:
- [0021] 当所述更新后的规则代码在执行过程中出现异常时,获取更新前的规则代码对应的可用版本;
- [0022] 从当前使用版本回退至更新前的规则代码对应的可用版本。
- [0023] 在其中一个实施例中,所述方法还包括:
- [0024] 当接收到业务请求时,根据所述业务请求执行更新后的规则代码所对应的业务场景;
- [0025] 通过业务系统调用更新后的规则代码执行所述业务请求;
- [0026] 根据所述业务方请求中包含的场景标识,规则标识匹配到终端提供的相应的业务场景调用接口;
- [0027] 当所述业务方调用并执行相应的业务场景时,所述业务系统通过规则引擎解析相应的规则代码,执行所述规则对应的业务场景;
- [0028] 对所述业务场景所对应的规则调用次数进行统计。
- [0029] 一种业务规则更新装置,所述装置包括:
- [0030] 接收模块,用于当监听到分布式服务组件接收到的规则版本发生变化时,接收所述分布式服务组件发出的业务场景下至少一个规则对应的规则更新消息;
- [0031] 更新模块,用于根据所述规则更新消息获取更新后的规则所对应的更新后的规则代码,将所述更新后的规则代码缓存到本地磁盘;
- [0032] 调用模块,用于利用所述更新后的规则代码对内存进行更新,通过在内存中调用所述更新后的规则代码执行更新后的规则。
- [0033] 在其中一个实施例中,所述接收模块还用于通过页面代码编辑器接收更新后的规则代码,存入数据库;所述更新模块还用于将所述更新后的规则代码对应的规则版本设置为当前使用版本;将所述Zookeeper中更新前的规则版本变更为所述当前使用版本。
- [0034] 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以实现以下步骤:
- [0035] 当监听到分布式服务组件接收到的规则版本发生变化时,接收所述分布式服务组件发出的业务场景下至少一个规则对应的规则更新消息;
- [0036] 根据所述规则更新消息获取更新后的规则所对应的更新后的规则代码,将所述更新后的规则代码缓存到数据库;
- [0037] 利用所述更新后的规则代码对内存进行更新,通过在内存中调用所述更新后的规则代码执行更新后的规则。
- [0038] 一种计算机可读存储介质,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以下步骤:
- [0039] 当监听到分布式服务组件接收到的规则版本发生变化时,接收所述分布式服务组件发出的业务场景下至少一个规则对应的规则更新消息;
- [0040] 根据所述规则更新消息获取更新后的规则所对应的更新后的规则代码,将所述更新后的规则代码缓存到数据库;
- [0041] 利用所述更新后的规则代码对内存进行更新,通过在内存中调用所述更新后的规

则代码执行更新后的规则。

[0042] 上述规则更新方法、装置、计算机设备和存储介质,通过对分布式服务组件接收的数据进行监听,当监听到分布式服务组件接收到的规则发生变化时,接收所述分布式服务组件发出的业务场景下至少一个规则对应的规则更新消息。如果业务场景下有多条规则更新,可以接收到多个业务规则更新消息。根据所述规则更新消息获取更新后的规则代码,将所述更新后的规则代码缓存到本地磁盘。通过在内存中调用所述更新后的规则代码执行更新后的规则。由此,通过分布式服务组件来通知场景标识、规则标识的变更到业务系统,这样使得规则的变化全都由系统触发而自动更新,从而快速实现了业务系统中规则代码的自动更新,提高了规则的更新效率。

### 附图说明

- [0043] 图1为一个实施例中规则更新方法的应用环境图;
- [0044] 图2为一个实施例中规则更新方法的流程图;
- [0045] 图3为一个实施例中规则更新过程的示意图;
- [0046] 图4为一个实施例中规则更新装置的结构示意图;
- [0047] 图5为另一个实施例中规则更新装置的结构示意;
- [0048] 图6为一个实施例中计算机设备的内部结构图。

### 具体实施方式

[0049] 为了使本申请的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本申请进行进一步详细说明。应当理解,此处描述的具体实施例仅仅用以解释本申请,并不用于限定本申请。

[0050] 本申请提供的业务规则更新方法,可以应用于如图1所示的应用环境中。其中,终端102通过网络与服务器104通过网络进行通信。服务器104包括业务系统、分布式服务组件和消息中间件,分布式服务组件包括Zookeeper。业务系统与分布式服务组件保存长链接,并监听分布式服务组件中的数据变化,当监听到分布式服务组件中规则版本发生变化时,分布式服务组件将规则更新消息发送到业务系统。服务器104接收终端102发送的规则更新消息,并根据规则更新消息更新分布式服务组件中对应的场景标识、规则所对应的规则版本。分布式服务组件中有业务场景下规则版本发生变化,业务系统接收到分布式服务组件发送的规则更新消息,根据规则更新消息获取场景标识,规则所对应的规则版本,通过http (HyperText Transfer Protocol,超文本传输协议) 请求获取终端102在页面编辑器更新后的规则代码。服务器104执行更新后的规则代码。实现了规则的自动更新。其中,终端102可以但不限于各种个人计算机、笔记本电脑、智能手机、平板电脑和便携式可穿戴设备,服务器104可以用独立的服务器或者是多个服务器组成的服务器集群来实现。

[0051] 在一个实施例中,如图2所示,提供了一种规则更新方法,以该方法应用于图1中的服务器为例进行说明,具体包括:

[0052] 步骤202,当监听到分布式服务组件接收到的规则版本发生变化时,接收分布式服务组件发出的业务场景下至少一个规则对应的规则更新消息。

[0053] 业务系统对分布式服务组件中的数据变化进行监听。当分布式服务组件接收到终

端规则更新消息时,分布式服务组件获取规则更新消息中的场景标识、规则标识,根据场景标识将相应业务场景下规则标识对应的规则版本更新为规则更新消息中规则标识对应的规则版本。当业务系统监听到分布式服务组件中的规则版本发生变化时,分布式服务组件把业务场景下至少一个业务规则对应的规则更新消息发送至业务系统。

[0054] 步骤204,根据规则更新消息获取更新后的规则所对应的更新后的规则代码,将更新后的规则代码缓存到本地磁盘。

[0055] 分布式服务组件将规则更新消息发送到业务系统,业务系统获取规则更新消息中的场景标识、规则标识,根据场景标识、规则标识获取对应的更新后的规则版本。根据获取到的更新后的规则版本通过http请求从终端更新后的规则代码所对应的代码块中获取更新后的规则代码。

[0056] 步骤206,利用更新后的规则代码对内存进行更新,通过在内存中调用更新后的规则代码执行更新后的规则。

[0057] 当需要执行业务规则时,通过获取业务方的业务请求数据中包含的场景标识、规则标识,调用终端提供的相应接口就可执行相应的业务场景。当代码执行到对应的业务场景的时候,业务系统会通过规则引擎解析内存中保存的更新后的规则代码,并执行相应业务场景下更新后的规则代码。

[0058] 在传统的方式,业务规则更新时,需要通过开发人员编写代码,人工对代码进行编译、测试以及发布。所有工作流程均由人工操作完成,给开发人员增加了不必要的工作,降低了业务规则的更新效率。

[0059] 本实施例中,通过业务系统监听分布式服务组件的数据变化,当分布式服务组件中业务规则发生变化时,发送业务规则更新消息到业务系统,业务系统根据规则更新消息中的最新版本号通过http请求从规则平台管理后台获取最新的规则代码。并把最新规则代码保存到业务系统所在的本地磁盘中,同时更新业务系统所占用的内存中对应的规则代码。由此,通过规则更新消息的传送,使得业务系统自动更新规则代码到业务系统所在的本地磁盘中,并同时更新业务系统所占用的内存中对应的规则代码。节省了不必要的人工操作环节,提高业务规则更新效率。

[0060] 在一个实施例中,分布式服务组件包括Zookeeper,该方法还包括:通过页面代码编辑器接收更新后的规则代码,存入数据库;将更新后的规则代码对应的规则版本设置为当前使用版本;将Zookeeper中更新前的规则版本变更为当前使用版本。

[0061] 本实施例中,规则代码为已有代码,终端选择规则场景下需要修改更新的对应的规则代码,通过页面在线代码编辑器修改成相应业务逻辑的规则代码,将更新后的规则代码保存至数据库,同时生成新的规则版本并保存到数据库中。将更新后的规则代码对应的规则版本设置为当前使用版本,将Zookeeper中保存的更新前的规则版本变更为当前使用版本。通过页面代码编辑器编辑修改规则代码将更新的规则代码保存生成新版本,并保存到数据库,将新版本设置为当前使用版本后,将更新后的规则代码所对应的规则版本同步到Zookeeper中。由此,通过页面代码编辑器接收更新后的规则代码,将更新后的规则代码对应的新版本设置为当前使用版本,并同时更新Zookeeper中对应的规则版本信息,实现了规则版本信息的同步更新。

[0062] 在另一个实施例中,规则代码为新增代码,终端通过新增创建一个业务场景,在业

务场景下通过新增创建规则。其中,一个业务场景包含场景编码、场景名称、场景说明、规则等,一条规则包含规则编码、规则名称、规则说明、代码块等,代码块的填写采用CodeMirror(在线代码编辑器)编写。进一步的,一个业务场景可以支持多个相同逻辑的规则,并对多个规则进行优先级的排序。由此,通过对一个业务场景下的多个规则进行优先级的排序,实现了多个规则按优先级高低的合理使用。

[0063] 在一个实施例中,通过页面代码编辑器接收更新后的规则代码后,该方法还包括:接收测试请求,对测试请求返回测试页面;接收测试页面中的测试参数,执行测试请求;当执行测试请求的返回值和预期结果一致,保存更新后的规则代码至数据库。

[0064] 本实施例中,终端获取服务器返回的测试页面,在测试页面中输入测试参数,测试参数包括参数名和参数值等。服务器获取测试页面中输入的测试参数,根据测试参数执行相应的测试请求。服务器将测试结果返回测试页面,当测试页面中测试结果的返回值与预期结果一致,将更新后的规则代码保存到数据库。如果测试结果与预期结果不一致,则重新修改规则代码并进行测试。由此,通过服务器获取测试请求并测试、验证规则代码的执行结果,节省了不必要的工作环节,提高了规则代码更新后的测试效率,保证了规则代码更新后的准确性。

[0065] 在另一个实施例中,还可以对规则代码进行批量测试,打开单元测试参数页面,填写批量测试的参数名、参数值、返回值。填写好批量测试所需的测试参数后执行测试,在单元测试参数页面中输出所有执行情况,当所有数据通过测试时,将更新后的规则代码保存至数据库。如果测试不通过,分析测试结果,定位出错的规则代码,并修改重测。由此,通过服务器进行批量测试,验证更新后的规则代码的执行结果,节省了不必要的工作,提高了规则代码的测试效率,保证规则代码更新后的准确率。

[0066] 在一个实施例中,根据规则更新消息获取更新后的规则所对应的更新后的规则代码包括:在规则更新消息中提取与更新后的规则对应的场景标识、版本标识;版本标识为当前使用版本;根据场景标识、当前使用版本获取更新后的规则代码。

[0067] 本实施例中,业务系统监听Zookeeper中的数据变化情况,当Zookeeper有规则版本更新变化时,Zookeeper将规则更新消息发送到业务系统,规则更新消息包含有场景标识、规则对应的版本标识,业务系统根据更新消息包含的场景标识、规则对应的版本标识获取更新后的规则版本通过http请求方式从终端获取更新后的规则代码。业务系统将获取到的规则更新消息对应的规则编码和当前使用版本对应的更新后的规则代码保存到业务系统所在的本地磁盘,并同时更新业务系统所占用的内存中对应的规则代码。由此,通过获取规则更新消息包含的场景标识、规则对应的版本标识来获取更新后的规则代码,并保存到业务系统对应的本地磁盘和内存中,通过Zookeeper来通知规则标识、场景标识的变更到各业务系统,使得规则的变化都由系统触发而自动更新,节省人工操作环节,提高了规则更新的效率。

[0068] 以业务场景为考试分数转换为例对上述实施例进行说明。如图3所示,图3中包括更新前的规则代码302、更新后的规则代码304、更新前的规则版本信息306、更新后的规则版本信息308。其中,更新前的版本信息包括,场景编码:scene\_score2chinese,规则编码:rule\_score2chinese,规则版本:rule\_2019070200000,当前使用版本:rule\_2019070200000。更新后的版本信息包括,场景编码:scene\_score2chinese,规则编码:

rule\_score2chinese,规则版本:rule\_2019071600000,当前使用版本:rule\_2019071600000。

[0069] 具体的,将上述规则代码302修改后保存生成更新后的规则代码304,更新后的规则代码304生成新的规则版本rule\_2019071600000,将修改后的规则代码保存到数据库。终端将新的规则版本rule\_2019071600000设置为当前使用版本时,Zookeeper下路径为/scene/scene\_score2chinese/rule\_score2chinese的值由当前的rule\_2019070200000修改为rule\_2019071600000。由于Zookeeper中的当前使用版本的数据有变化,则监听Zookeeper数据变化的业务系统将会收到规则更新消息通知。业务系统根据规则更新消息中包含的场景标识、版本标识,从Zookeeper中获取场景编码为scene\_score2chinese下规则编码为rule\_score2chinese对应的最新版本rule\_2019071600000,根据最新版本通过http请求从终端获取最新的规则代码。之后业务系统会将规则编码rule\_score2chinese和当前使用版本rule\_2019071600000的规则代码保存到业务系统所在的本地磁盘中,并同时更新业务系统所占用的内存中对应的规则代码。规则更新成功后,当执行规则时,执行的就是更新后的规则代码。

[0070] 在另一个实施例中,业务系统规则代码的更新,还可以通过消息队列与http接口调用的方式实现。进入终端,选择规则场景下需要修改更新的对应的规则代码,通过页面在线代码编辑器编辑成需要的规则代码,把更新后的规则代码保存至数据库,同时生成新的规则版本并保存到数据库中。将更新后的规则代码对应的规则版本设置为当前使用版本。终端将规则更新消息发送到消息中间件,业务系统监听到消息中间件有新的消息,并通过消息中间件获取规则更新消息,从规则更新消息中提取场景标识,版本标识。根据场景标识,版本标识获取对应的规则版本,根据获取的规则版本通过http请求方式从终端获取规则版本对应的更新后的规则代码。业务系统将获取到的规则更新消息中规则标识对应的规则编码和当前使用版本对应的更新后的规则代码保存到业务系统所在的本地磁盘,并同时更新业务系统所占用的内存中对应的规则代码。由此,通过获取规则更新消息包含的场景标识、版本标识来获取更新后的规则代码,并保存到业务系统对应的本地磁盘和内存中,通过消息中间件接收终端发送的规则更新消息,并把规则更新消息转发到业务系统,使得规则的更新变化全部由系统触发而自动更新,节省人工操作环节,提高了规则更新的效率。

[0071] 在一个实施例中,当更新后的规则代码在执行过程中出现异常时,获取更新前的规则代码对应的可用版本;从当前使用版本回退至更新前的规则代码对应的可用版本。

[0072] 本实施例中,当业务系统在执行更新后的规则代码出现异常时,可在终端管理的规则版本中将更新前的可用版本设置为当前使用版本。进一步的,更新前的可用版本可以是多个历史版本中的任一可用版本。由此,通过规则的多版本控制,防止了编写规则出错只能再次修改的情况,实现了规则版本的灵活控制。

[0073] 在一个实施例中,当接收到业务请求时,根据业务请求执行更新后的规则代码所对应的业务场景;通过业务系统调用更新后的规则代码执行业务请求;根据业务方请求中包含的场景标识,规则标识匹配到终端提供的相应的业务场景调用接口;当业务方调用并执行相应的业务场景时,业务系统通过规则引擎解析相应的规则代码,执行规则对应的业务场景;对业务场景所对应的规则调用次数进行统计。

[0074] 本实施例中,当接收到业务方发送的业务请求时,根据业务请求中包含的场景标

识,规则标识,匹配到终端提供的相应的业务场景的调用接口。当业务方调用到并执行相应的业务场景时,业务系统通过规则引擎解析相应的规则代码,执行业务场景下对应的规则代码。由此,通过规则引擎解析规则代码,使得业务方能够调用接口执行相应的场景。业务方请求调用成功后,终端对每个业务场景和业务场景所对应的规则的执行情况,保存到数据库中,对数据库中保存的数据进行统计。由此,通过对业务场景所对应的规则调用次数进行统计,方便终端对规则的管理。

[0075] 在一个实施例中,如图4所示,提供了一种规则更新装置,包括:接收模块402,更新模块404,调用模块406,其中:

[0076] 接收模块402,用于当监听到分布式服务组件接收到的规则版本发生变化时,接收分布式服务组件发出的业务场景下至少一个规则对应的规则更新消息。

[0077] 更新模块404,用于根据规则更新消息获取更新后的规则所对应的更新后的规则代码,将更新后的规则代码缓存到数据库。

[0078] 调用模块406,用于利用更新后的规则代码对内存进行更新,通过在内存中调用更新后的规则代码执行更新后的规则。

[0079] 在一个实施例中,接收模块402还用于通过页面代码编辑器接收更新后的规则代码,存入数据库;更新模块404还用于将更新后的规则代码对应的规则版本设置为当前使用版本;将Zookeeper中更新前的规则版本变更为当前使用版本。

[0080] 在一个实施例中,接收模块402还用于接收测试请求,对测试请求返回测试页面;接收测试页面中的测试参数,执行测试请求;更新模块404,还用于当执行测试请求的返回值和预期结果一致,保存更新后的规则代码至数据库。

[0081] 在一个实施例中,接收模块402,还用于在规则更新消息中提取与更新后的规则对应的场景标识、版本标识;版本标识为当前使用版本;根据场景标识、当前使用版本获取更新后的规则代码。

[0082] 在一个实施例中,接收模块402,还用于当更新后的规则代码在执行过程中出现异常时,获取更新前的规则代码对应的可用版本;从当前使用版本回退至更新前的规则代码对应的可用版本。

[0083] 在一个实施例中,如图5所示,接收模块502还用于当接收到业务请求时,根据业务请求执行更新后的规则代码所对应的业务场景;调用模块506还用于通过业务系统调用更新后的规则代码执行业务请求;根据业务方请求中包含的场景标识,规则标识匹配到终端提供的相应的业务场景调用接口;该装置还包括:

[0084] 执行模块508,用于当业务方调用并执行相应的业务场景时,业务系统通过规则引擎解析相应的规则代码,执行规则代码所对应的业务场景;统计模块510,用于对业务场景所对应的规则调用次数进行统计。

[0085] 在一个实施例中,提供了一种计算机设备,其内部结构图可以如图6所示。该计算机设备包括通过系统总线连接的处理器、存储器、网络接口。其中,该计算机设备的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质、内存储器。该非易失性存储介质存储有操作系统和计算机程序。该内存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该计算机设备的网络接口用于与外部的终端通过网络连接通信。该计算机程序被处理器执行时以实现一种业务规则更新方法。

[0086] 在一个实施例中,提供了一种计算机可读存储介质,其上存储有计算机程序,计算机程序被处理器执行时实现上述各个实施例提供的业务规则更新方法中的步骤。

[0087] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是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非易失性计算机可读存储介质中,该计算机程序在执行时,可包括如上述各方法的实施例的流程。其中,本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用,均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器 (ROM)、可编程ROM (PROM)、电可编程ROM (EPROM)、电可擦除可编程ROM (EEPROM) 或闪存。易失性存储器可包括随机存取存储器 (RAM) 或者外部高速缓冲存储器。作为说明而非局限,RAM以多种形式可得,诸如静态RAM (SRAM)、动态RAM (DRAM)、同步DRAM (SDRAM)、双数据率SDRAM (DDRSDRAM)、增强型SDRAM (ESDRAM)、同步链路 (Synchlink) DRAM (SLDRAM)、存储器总线 (Rambus) 直接RAM (RDRAM)、直接存储器总线动态RAM (DRDRAM)、以及存储器总线动态RAM (RDRAM) 等。

[0088] 以上实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。

[0089] 以上所述实施例仅表达了本申请的几种实施方式,其描述较为具体和详细,但不能因此而理解为对发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本申请构思的前提下,还可以做出若干变形和改进,这些都属于本申请的保护范围。因此,本申请专利的保护范围应以所附权利要求为准。

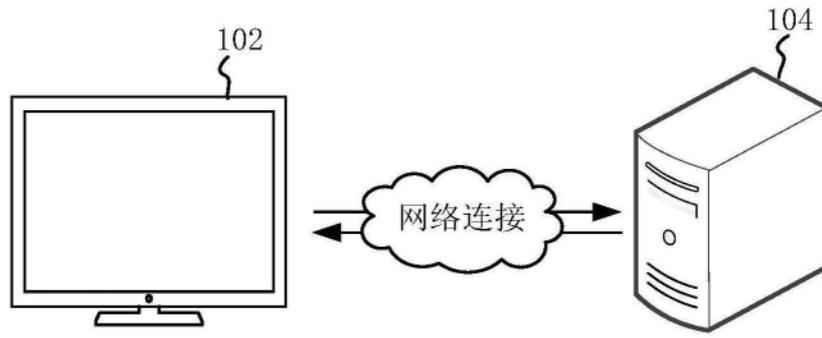


图1

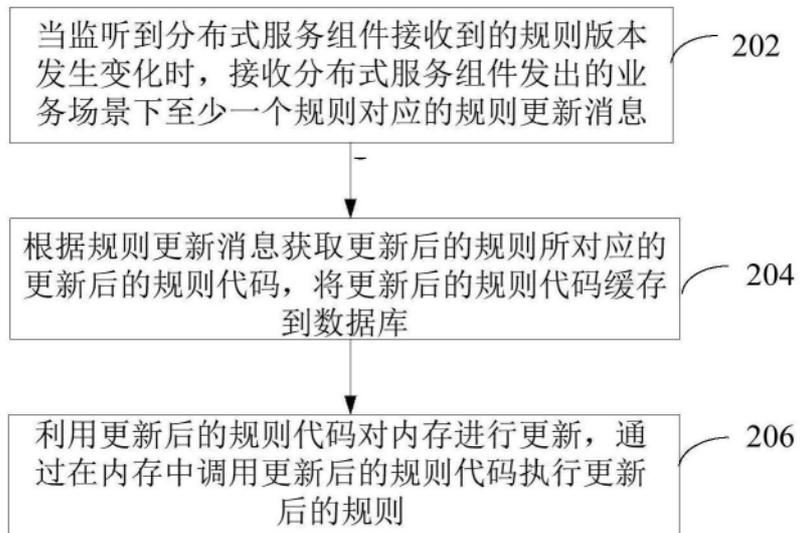


图2

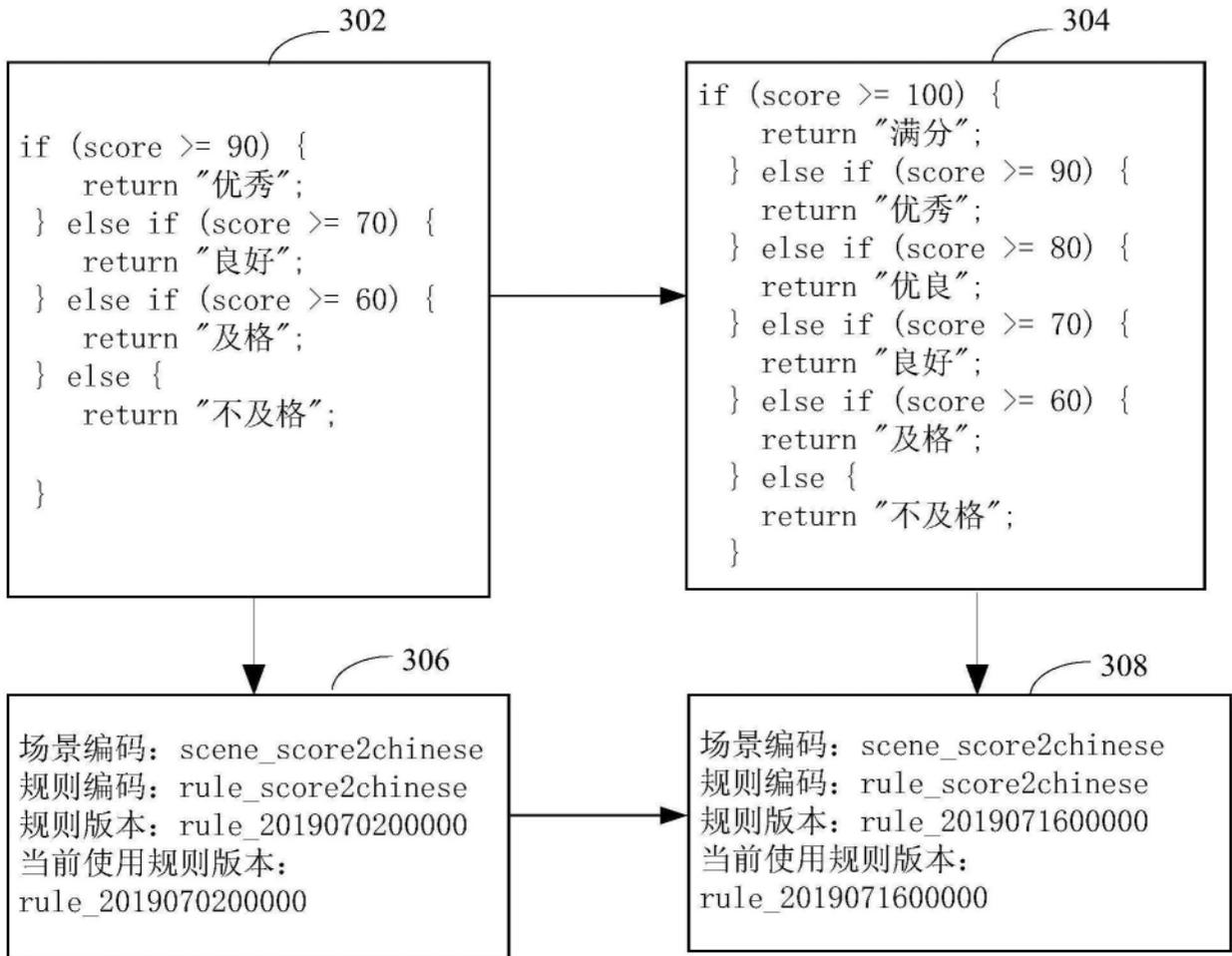


图3

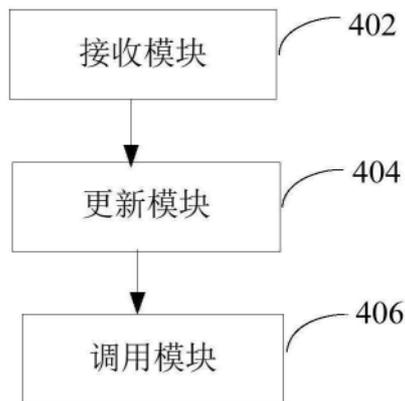


图4

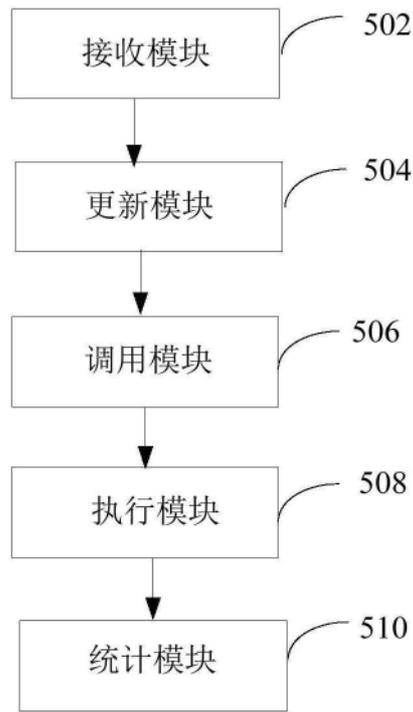


图5

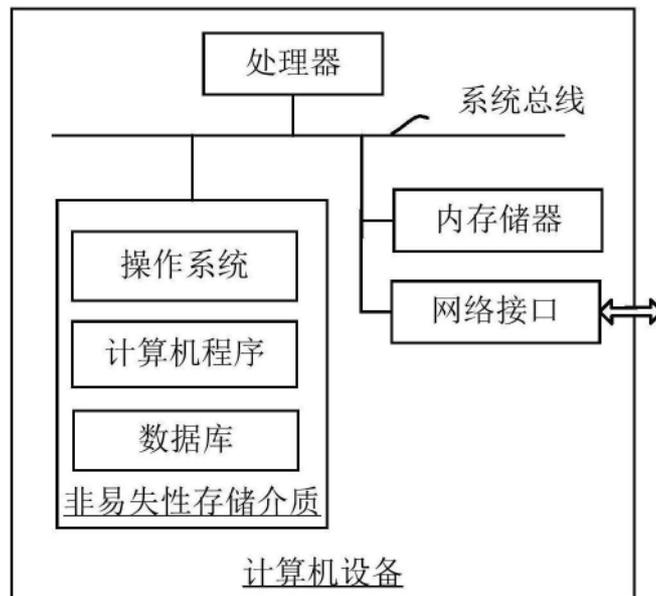


图6