



(12) 发明专利申请

(10) 申请公布号 CN 112995304 A

(43) 申请公布日 2021.06.18

(21) 申请号 202110171168.9

(22) 申请日 2021.02.08

(71) 申请人 中国工商银行股份有限公司  
地址 100140 北京市西城区复兴门内大街  
55号

(72) 发明人 程冲 滕达 张丹枫 唐冬雪

(74) 专利代理机构 北京三友知识产权代理有限公司 11127  
代理人 任默闻 孙乳笋

(51) Int.Cl.  
H04L 29/08 (2006.01)

权利要求书2页 说明书12页 附图8页

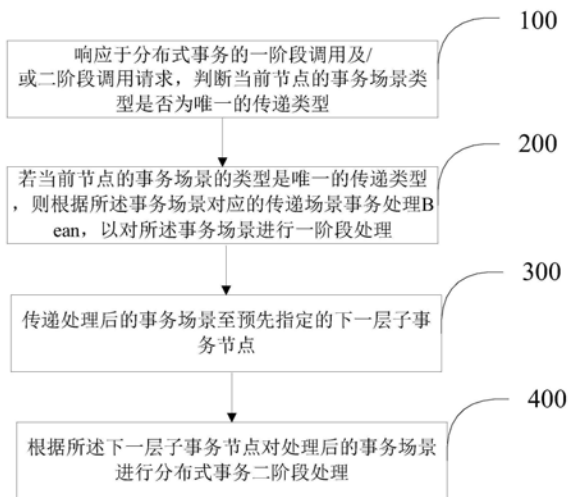
(54) 发明名称

二阶段分布式事务对于路由服务节点的处理方法及装置

(57) 摘要

本发明属于涉及大数据技术领域,本发明提供了一种二阶段分布式事务对于路由服务节点的处理方法及装置,二阶段分布式事务对于路由服务节点的处理方法包括:响应于分布式事务的一阶段调用及/或二阶段调用请求,判断当前节点的事务场景类型是否为唯一的传递类型;若当前节点的事务场景的类型是唯一的传递类型,则根据所述事务场景对应的传递场景事务处理Bean,以对所述事务场景进行一阶段处理;传递处理后的事务场景至预先指定的下一层子事务节点;根据所述下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。本发明消除路由服务节点的数据源,并且支持路由服务节点的负载均衡机制。

CN 112995304 A



1. 一种二阶段分布式事务对于路由服务节点的处理方法,其特征在于,包括:  
响应于分布式事务的一阶段调用及/或二阶段调用请求,判断当前节点的事务场景的类型是否为唯一的传递类型;

若当前节点的事务场景的类型是唯一的传递类型,则根据所述事务场景对应的传递场景事务处理Bean,以对所述事务场景进行一阶段处理;

传递处理后的事务场景至预先指定的下一层子事务节点;

根据所述下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。

2. 如权利要求1所述的二阶段分布式事务对于路由服务节点的处理方法,其特征在于,所述判断事务场景类型是否为唯一的传递类型包括:

根据配置值确定所述当前节点的事务场景类型。

3. 如权利要求1所述的二阶段分布式事务对于路由服务节点的处理方法,其特征在于,所述根据所述事务场景对应的传递场景事务处理Bean,以对所述事务场景进行一阶段处理,包括:

利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;

将所述主事务ID以及所述子事务ID写入事务链路的第一上下文中;

判断当前节点的事务场景是否为传递类型;

若当前节点的事务场景为传递类型,则根据所述第一上下文以及当前节点在一阶段处理中指定的下一层子事务节点所对应的一阶段方法处理所述事务场景;

若当前节点的事务场景不为传递类型,则根据事务数据库以及当前节点一阶段方法中制定的下一层子事务节点对应的一阶段方法处理所述事务场景。

4. 如权利要求1所述的二阶段分布式事务对于路由服务节点的处理方法,其特征在于,根据所述下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理包括:

利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;

将所述主事务ID和子事务ID写入事务链路的第二上下文中;

判断当前节点的事务场景是否为传递类型;

若当前节点的事务场景为传递类型,则根据所述第二上下文以及当前节点在二阶段处理中指定的下一层子事务节点所对应的二阶段方法处理所述事务场景;

若当前节点的事务场景不为传递类型,则查询所述事务数据库,以获取所述当前节点的下一层子事务ID以及对应的方法,并对所述事务场景进行处理。

5. 一种二阶段分布式事务对于路由服务节点的处理装置,其特征在于,包括:

类型判断模块,用于响应于分布式事务的一阶段调用及/或二阶段调用请求,判断当前节点的事务场景类型是否为唯一的传递类型;

一阶段处理模块,用于若当前节点的事务场景的类型是唯一的传递类型,则根据所述事务场景对应的传递场景事务处理Bean,以对所述事务场景进行一阶段处理;

事务场景传递模块,用于传递处理后的事务场景至预先指定的下一层子事务节点;

二阶段处理模块,用于根据所述下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。

6. 如权利要求5所述的二阶段分布式事务对于路由服务节点的处理装置,其特征在于,所述类型判断模块具体用于根据配置值确定所述当前节点的事务场景类型。

7. 如权利要求5所述的二阶段分布式事务对于路由服务节点的处理装置,其特征在于,所述一阶段处理模块包括:

一阶段ID确定单元,用于利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;

第一上下文写入单元,用于将所述主事务ID以及所述子事务ID写入事务链路的第一上下文中;

一阶段判断单元,用于判断当前节点的事务场景是否为传递类型;

一阶段第一处理单元,用于若当前节点的事务场景为传递类型,则根据所述第一上下文以及当前节点在一阶段处理中指定的下一层子事务节点所对应的一阶段方法处理所述事务场景;

一阶段第二处理单元,用于若当前节点的事务场景不为传递类型,则根据事务数据库以及当前节点一阶段方法中制定的下一层子事务节点对应的一阶段方法处理所述事务场景。

8. 如权利要求5所述的二阶段分布式事务对于路由服务节点的处理装置,其特征在于,所述二阶段处理模块包括:

二阶段ID确定单元,用于利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;

第二上下文写入单元,用于将所述主事务ID和子事务ID写入事务链路的第二上下文中;

二阶段判断单元,用于判断当前节点的事务场景是否为传递类型;

二阶段第一处理单元,用于若当前节点的事务场景为传递类型,则根据所述第二上下文以及当前节点在二阶段处理中指定的下一层子事务节点所对应的二阶段方法处理所述事务场景;

二阶段第二处理单元,用于若当前节点的事务场景不为传递类型,则查询所述事务数据库,以获取所述当前节点的下一层子事务ID以及对应的方法,并对所述事务场景进行处理。

9. 一种电子设备,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程序时实现权利要求1至4任一项所述二阶段分布式事务对于路由服务节点的处理方法的步骤。

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,该计算机程序被处理器执行时实现权利要求1至4任一项所述二阶段分布式事务对于路由服务节点的处理方法的步骤。

## 二阶段分布式事务对于路由服务节点的处理方法及装置

### 技术领域

[0001] 本申请涉及大数据技术领域,具体涉及一种二阶段分布式事务对于路由服务节点的处理方法及装置。

### 背景技术

[0002] 针对高并发、大交易量的业务场景,业界往往是将部分应用拆分出路由服务节点,该节点具有以下特点:无业务逻辑,仅提供负载均衡能力,支持多种负载均衡算法(使用较多的算法是随机路由)。在路由服务节点中,根据请求参数的公共业务要素(类似账务服务的账号),按照分片计算规则,将请求转发各多个业务组合服务节点。外围应用,统一调用路由服务节点,由路由服务节点选择具体的业务组合服务节点。

[0003] 在两阶段的分布式事务中,事务的调用链路中,可能存在路由服务节点。路由服务节点无业务逻辑,无需配置数据源,且对于调用请求通过负载均衡算法进行路由。对于两阶段事务模式,事务的一阶段和二阶段操作,可能负载到不同的节点。二阶段的操作中,调用到路由服务节点与一阶段不同,无法获取到下一层子事务信息,导致调用失败。

[0004] 综上所述,如何解决调用方和路由服务节点,以及路由服务节点与下一层的业务组合节点的事务信息存储和二阶段操作的问题是亟需解决的问题。

### 发明内容

[0005] 本发明属于大数据技术领域,本发明所提供的二阶段分布式事务对于路由服务节点的处理方法及装置,消除了路由服务节点的数据源,支持路由服务节点的负载均衡机制。其优点如下:

[0006] 1.解耦数据源。基于Spring上下文的条件加载机制,对于传递类型的事务场景,屏蔽数据库操作相关的bean。

[0007] 2.支持路由服务节点的负载均衡能力,通过RpcContext的Filter机制,设置二阶段操作标识,显式的调用下一层子事务的二阶段操作。

[0008] 为解决上述技术问题,本发明提供以下技术方案:

[0009] 响应于分布式事务的一阶段调用及/或二阶段调用请求,判断当前节点的事务场景类型是否为唯一的传递类型;

[0010] 若当前节点的事务场景的类型是唯一的传递类型,则根据所述事务场景对应的传递场景事务处理Bean,以对所述事务场景进行一阶段处理;

[0011] 传递处理后的事务场景至预先指定的下一层子事务节点;

[0012] 根据所述下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。

[0013] 一实施例中,所述判断事务场景类型是否为唯一的传递类型包括:

[0014] 根据配置值确定所述当前节点的事务场景类型。

[0015] 一实施例中,所述根据所述事务场景对应的传递场景事务处理Bean,以对所述事务场景进行一阶段处理,包括:

- [0016] 利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;
- [0017] 将所述主事务ID以及所述子事务ID写入事务链路的第一上下文中;
- [0018] 判断当前节点的事务场景是否为传递类型;
- [0019] 若当前节点的事务场景为传递类型,则根据所述第一上下文以及当前节点在一阶段处理中指定的下一层子事务节点所对应的一阶段方法处理所述事务场景;
- [0020] 若当前节点的事务场景不为传递类型,则根据事务数据库以及当前节点一阶段方法中制定的下一层子事务节点对应的一阶段方法处理所述事务场景。
- [0021] 一实施例中,根据所述下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理包括:
- [0022] 利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;
- [0023] 将所述主事务ID和子事务ID写入事务链路的第二上下文中;
- [0024] 判断当前节点的事务场景是否为传递类型;
- [0025] 若当前节点的事务场景为传递类型,则根据所述第二上下文以及当前节点在二阶段处理中指定的下一层子事务节点所对应的二阶段方法处理所述事务场景;
- [0026] 若当前节点的事务场景不为传递类型,则查询所述事务数据库,以获取所述当前节点的下一层子事务ID以及对应的方法,并对所述事务场景进行处理。
- [0027] 第二方面,本发明提供一种二阶段分布式事务对于路由服务节点的处理装置,该装置包括:
- [0028] 类型判断模块,用于响应于分布式事务的一阶段调用及/或二阶段调用请求,判断当前节点的事务场景类型是否为唯一的传递类型;
- [0029] 一阶段处理模块,用于若当前节点的事务场景的类型是唯一的传递类型,则根据所述事务场景对应的传递场景事务处理Bean,以对所述事务场景进行一阶段处理;
- [0030] 事务场景传递模块,用于传递处理后的事务场景至预先指定的下一层子事务节点;
- [0031] 二阶段处理模块,用于根据所述下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。
- [0032] 一实施例中,所述类型判断模块具体用于根据配置值确定所述当前节点的事务场景类型。
- [0033] 一实施例中,所述一阶段处理模块包括:
- [0034] 一阶段ID确定单元,用于利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;
- [0035] 第一上下文写入单元,用于将所述主事务ID以及所述子事务ID写入事务链路的第一上下文中;
- [0036] 一阶段判断单元,用于判断当前节点的事务场景是否为传递类型;
- [0037] 一阶段第一处理单元,用于若当前节点的事务场景为传递类型,则根据所述第一上下文以及当前节点在一阶段处理中指定的下一层子事务节点所对应的一阶段方法处理所述事务场景;
- [0038] 一阶段第二处理单元,用于若当前节点的事务场景不为传递类型,则根据事务数据库以及当前节点一阶段方法中制定的下一层子事务节点对应的一阶段方法处理所述事

务场景。

[0039] 一实施例中,所述二阶段处理模块包括:

[0040] 二阶段ID确定单元,用于利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;

[0041] 第二上下文写入单元,用于将所述主事务ID和子事务ID写入事务链路的第二上下文中;

[0042] 二阶段判断单元,用于判断当前节点的事务场景是否为传递类型;

[0043] 二阶段第一处理单元,用于若当前节点的事务场景为传递类型,则根据所述第二上下文以及当前节点在二阶段处理中指定的下一层子事务节点所对应的二阶段方法处理所述事务场景;

[0044] 二阶段第二处理单元,用于若当前节点的事务场景不为传递类型,则查询所述事务数据库,以获取所述当前节点的下一层子事务ID以及对应的方法,并对所述事务场景进行处理。

[0045] 第三方面,本发明提供一种电子设备,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,处理器执行程序时实现二阶段分布式事务对于路由服务节点的处理方法的步骤。

[0046] 第四方面,本发明提供一种计算机可读存储介质,其上存储有计算机程序,该计算机程序被处理器执行时实现二阶段分布式事务对于路由服务节点的处理方法的步骤。

[0047] 从上述描述可知,本发明实施例提供一种二阶段分布式事务对于路由服务节点的处理方法及装置,首先响应于分布式事务的一阶段调用及/或二阶段调用请求,判断当前节点的事务场景类型是否为唯一的传递类型;若是,则根据事务场景对应的传递场景事务处理Bean,以对事务场景进行一阶段处理;接着,传递处理后的事务场景至预先指定的下一层子事务节点;最后下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。本发明通过Spring上下文的条件加载机制,新增事务场景TCC\_TRANSMIT(传递类型),在启动时,数据库操作相关的bean(java类)不进行加载;通过分布式服务Dubbo协议的Filter机制,将路由服务节点和业务服务节点作为一个整体进行处理,路由服务节点无需存储下一层子事务的事务信息,通过二阶段操作显式的调用业务服务节点,完成二阶段操作的调用。本发明消除路由服务节点的数据源,并且支持路由服务节点的负载均衡机制,具体地,具有有益效果如下:

[0048] 1. 解耦数据源:基本Spring上下文的条件加载机制,对于传递类型的事务场景,屏蔽数据库操作相关的bean。

[0049] 2. 支持路由服务节点的负载均衡能力,通过RpcContext的Filter机制,设置二阶段操作标识,显式的调用下一层子事务的二阶段操作。

## 附图说明

[0050] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术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。

- [0051] 图1为本发明实施例中二阶段分布式事务对于路由服务节点的处理方法的流程示意图；
- [0052] 图2为本发明实施例中步骤200的流程示意图一；
- [0053] 图3为本发明实施例中步骤400的流程示意图一；
- [0054] 图4为本发明实施例中步骤100的流程示意图；
- [0055] 图5为本发明实施例中步骤200的流程示意图二；
- [0056] 图6为本发明实施例中步骤400的流程示意图二；
- [0057] 图7为本发明具体应用实例中二阶段分布式事务对于路由服务节点的处理方法的时序图；
- [0058] 图8为本发明的实施例中二阶段分布式事务对于路由服务节点的处理装置的结构示意图；
- [0059] 图9为本发明的实施例中一阶段处理模块20的结构示意图；
- [0060] 图10为本发明的实施例中二阶段处理模块40的结构示意图；
- [0061] 图11为本发明的实施例中的电子设备的结构示意图。

### 具体实施方式

[0062] 为使本发明实施例的目的、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、完整的描述，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作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。

[0063] 本发明的实施例提供一种二阶段分布式事务对于路由服务节点的处理方法的具体实施方式，参见图1，其具体包括如下内容：

[0064] 步骤100：响应于分布式事务的一阶段调用及/或二阶段调用请求，判断当前节点的事务场景类型是否为唯一的传递类型。

[0065] 两阶段模型的事务信息传递和存储，现有技术中，解决方案包括以下两种：一种是定义事务信息的公共输入和输出。业务接口的入参和出参必须继承自公共输入和输出，侵入性大，破坏了开闭原则；另一种是通过数据库保存事务信息，在每一层事务配置对应的数据库，通过表存储下一层的事务信息。此解决方法，要求一阶段和二阶段，可以访问到同一个数据库。对于路由服务节点，对其架构侵入性较大。

[0066] 需要指出的是，这里的一阶段调用以及二阶段调用（二阶段分布式事务）是指在时间上将事务场景的处理分为两个阶段，具体地，一阶段是指服务节点至路由节点；二阶段是指路由节点至子事务节点。另外，可根据事务场景中的配置值判断该事务场景的具体类型。

[0067] 步骤200：若当前节点的事务场景的类型是唯一的传递类型，则根据所述事务场景对应的传递场景事务处理Bean，以对所述事务场景进行一阶段处理。

[0068] 可以理解的是，这里的Bean对应的是一种Java类，可以理解为一种具体的方法或者处理模型，具体地，Bean是描述Java的软件组件模型，类似于Microsoft的COM组件概念。在Java模型中，通过JavaBean可以无限扩充Java程序的功能，通过Bean的组合可以快速的生成新的应用程序。通过Bean可以实现代码的重复利用，另外对于程序的易维护性等等也

有很重大的意义。不同场景类型,封装不同的事务处理Bean,对应不同的处理流程和方法;

[0069] 另一方面,事务场景类型主要包括:普通场景和传递场景;普通类型的事务场景的事务发起方和参与方均需要设置对应的本地数据库,且本地数据库存在两种表:主事务表和子事务表。相对地,传递类型的事务场景只能作为参与方(主、子事务表都不需要配置),不需要配置数据源。传递类型的事务场景节点只能作为参与方。

[0070] 参见图2,校验事务场景是否为传递类型(TCC\_TRANSMIT)。如果是,判断场景否唯一。如果存在其他事务场景,则报错提示。如果事务场景类型为传递类型且唯一,则加载传递类型切面相关bean,数据库操作相关的bean不进行加载,以达到屏蔽数据库操作相关的bean。

[0071] 步骤300:传递处理后的事务场景至预先指定的下一层子事务节点。

[0072] 需要说明的是,如果事务场景为唯一的传递类型,则根据预先指定的路径将由步骤200处理后的事务场景发送至下一层子事务节点。

[0073] 步骤400:根据所述下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。

[0074] 参见图3,与一阶段处理一样,在步骤400中,首先判断事务场景是否为唯一的传递类型,如果是,则跳过查询当前数据库的事务表的处理(无需查询下一层子事务),直接执行二阶段方法(由于传递场景的事务,在二阶段无法感知下一层子事务,要求传递事务的二阶段方法中,显式调用下一层子事务的二阶段方法。执行传递事务的二阶段方法,则完成下一层子事务的二阶段方法调用)。

[0075] 从上述描述可知,本发明实施例提供一种二阶段分布式事务对于路由服务节点的处理方法及装置,通过Spring上下文的条件加载机制,新增事务场景TCC\_TRANSMIT(传递类型),在启动时,数据库操作相关的bean(java类)不进行加载;通过分布式服务Dubbo协议的Filter机制,将路由服务节点和业务服务节点作为一个整体进行处理,路由服务节点无需存储下一层子事务的事务信息,通过二阶段操作显式的调用业务服务节点,完成二阶段操作的调用。本发明消除路由服务节点的数据源,并且支持路由服务节点的负载均衡机制。

[0076] 一实施例中,参见图4,步骤100进一步包括:

[0077] 步骤101:根据配置值确定所述当前节点的事务场景类型。

[0078] 具体地,读取所述事务场景的配置值,接着,根据所述配置值判断所述事务场景类型是否为唯一的传递类型。在节点启动期间,将传递场景相关的事务处理Bean,加载到Spring上下文。获取该节点的所有事务服务,将一阶段方法和对应的二阶段方法,加载到Spring上下文。

[0079] 一实施例中,参见图5,步骤200进一步包括:

[0080] 步骤201:利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;

[0081] 步骤202:将所述主事务ID以及所述子事务ID写入事务链路的第一上下文中;

[0082] 步骤203:判断当前节点的事务场景是否为传递类型;

[0083] 判断方法与步骤101中的方法类似,这里不再累述。

[0084] 步骤204:若当前节点的事务场景为传递类型,则根据所述第一上下文以及当前节点在一阶段处理中指定的下一层子事务节点所对应的一阶段方法处理所述事务场景;



[0085] 步骤205:若当前节点的事务场景不为传递类型,则根据事务数据库以及当前节点一阶段方法中制定的下一层子事务节点对应的一阶段方法处理所述事务场景。

[0086] 在步骤201至步骤205中,利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;将所述主事务ID和子事务ID写入事务链路的上下文中;接着,判断当前阶段的事务场景是否为传递类型;如果是,则根据事务链路的上下文,直接发起当前节点一阶段方法中指定的下一层子事务节点的一阶段处理。如果不是,则根据事务链路的上下文,生成下一层的子事务ID,将主事务ID和下一层子事务ID,写入事务数据库存储。然后发起当前节点一阶段方法中指定的下一层子事务节点的一阶段处理。

[0087] 一实施例中,参见图6,步骤400进一步包括:

[0088] 步骤401:利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;

[0089] 步骤402:将所述主事务ID和子事务ID写入事务链路的第二上下文中;

[0090] 步骤403:判断当前节点的事务场景是否为传递类型;

[0091] 判断方法与步骤101中的方法类似,这里不再累述。

[0092] 步骤404:若当前节点的事务场景为传递类型,则根据所述第二上下文以及当前节点在二阶段处理中指定的下一层子事务节点所对应的二阶段方法处理所述事务场景;

[0093] 步骤405:若当前节点的事务场景不为传递类型,则查询所述事务数据库,以获取所述当前节点的下一层子事务ID以及对应的方法,并对所述事务场景进行处理。

[0094] 在步骤401至步骤405中,同样地利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;将所述主事务ID和子事务ID写入事务链路的上下文中;接着,判断当前阶段的事务场景是否为传递类型;如果是,则根据所述事务链路的上下文,直接发起当前节点二阶段方法中指定的下一层子事务节点的二阶段处理。如果不是,则根据事务链路的上下文,则根据主事务ID和上一层子事务ID,查询事务数据库,获取当前节点的下一层子事务ID和方法。根据查询结果,发起下一层子事务节点的二阶段处理。

[0095] 为进一步地说明本方案,本发明还提供二阶段分布式事务对于路由服务节点的处理方法的具体应用实例,具体包括以下步骤:

[0096] 为解决两阶段分布式事务模型中,路由服务节点未配置数据源、随机路由算法的事务信息传递和存储问题。本具体应用实例基于分布式服务调用的Filter机制(选择性的加载)和Spring上下文加载机制,提供了一种两阶段分布式事务支持路由服务节点负载均衡的方法。对于一阶段(服务节点到路由节点)处理有:

[0097] S1:对类型为传递类型的事务场景进行一阶段处理。

[0098] 可以理解的是,普通类型的事务场景的事务发起方和参与方,均需要设置对应的本地数据库,且本地数据库存在两种表:主事务表和子事务表。传递类型的事务场景的参与方(主、子事务表都不需要配置),不需要配置数据源。传递类型的事务场景节点只能作为参与方;

[0099] 主事务表仅在事务发起方写记录,记录整个主事务的执行情况。主要包含:主事务ID和主事务状态等字段。其中,主事务状态包含中间状态。子事务表在事务发起方和事务参与者,均会写记录,记录当前事务的下一层事务关系数据。主要包含:主事务ID、子事务ID、当前事务ID和子事务状态等字段。其中,主事务ID在整个事务调用链路中保持不变,子事务

ID则随着参与者的变化而不同,当前事务ID则表示子事所归属的上一层事务ID,可能为主事务ID也可能是上一层的子事务ID。

[0100] 子事务ID通过事务管理器,在分布式服务的消费方进行生成,同时写入消费方所在的数据库进行存储。子事务表,仅需要记录当前事务的直接子事务,不记录非直接子事务(例如:A调用B和C,C调用E和F。那么A会记录B、C的子事务信息,不记录E和F的子事务信息)。

[0101] 对于传递类型的事务场景,跳过当前数据库子事务表的处理(无需记录下一层子事务信息)。

[0102] 具体存储情况如表1至表4所示:主事务A,有3个子事务B、C和D。其中子事务D和主事务A属于同一个应用,子事务B是路由服务节点,调用嵌套的子事务E和F,子事务D调用嵌套的子事务G。主事务A和子事务D的数据库记录(表1和表4):

[0103] 表1主事务表

主 事 务 ID(GLOBAL_TX_ID)	状态
A1	CREATE

[0105] 表2子事务表

主 事 务 ID(GLOBAL_TX_ID)	子 事 务 ID (SUB_TX_ID)	当 前 事 务 ID(LOCAL_TX_ID)	状态
A1	B1	A1	CREATE
A1	C1	A1	CREATE
A1	D1	A1	CREATE
A1	G1	D1	CREATE

[0107] 注:表2中第四行表示子事务D1的直接子事务。

[0108] 对于子事务B,无需配置数据库。子事务C的数据库:

[0109] 表3主事务表

主 事 务 ID(GLOBAL_TX_ID)	状态

[0111] 表4子事务表

主 事 务 ID(GLOBAL_TX_ID)	子 事 务 ID (SUB_TX_ID)	当 前 事 务 ID(LOCAL_TX_ID)	状态

[0113] S2:传递一阶段处理后的事务场景至预先指定的下一层子事务节点。

[0114] S3:对类型为传递类型的事务场景进行二阶段处理。

[0115] 二阶段Confirm/Cancel,事务信息的传递和使用:在路由服务节点的二阶段操作中,必须显式(预先指定)的调用下一层的二阶段操作,具体地:

[0116] 参见图7,起方开始执行二阶段Confirm。发起方根据主事务ID和当前事务ID(此时

等于主事务ID), 查询本地数据库的子事务表, 获取当前的直接子事务列表B、C和D。发起方循环处理每一笔子事务的二阶段Confirm调用: 通过分布式服务的消费方, 通过消费方的Filter机制, 将表中主事务ID、子事务ID和二阶段操作标识(Confirm/Cancel), 写入RpcContext上下文中的LocalTxId。通过分布式服务的提供方Filter机制, 获取RpcContext上下文中的主事务ID、子事务ID和二阶段标识, 判断是否为TCC模型的二阶段调用。如果为TCC模型的二阶段调用, 则继续判断当前事务的场景是否为传递类型。如果是传递类型, 则跳过查询当前数据库的事务表的处理(无需查询下一层子事务), 直接执行二阶段方法(由于传递场景的事务, 在二阶段无法感知下一层子事务, 要求传递事务的二阶段方法中, 显式调用下一层子事务的二阶段方法。执行传递事务的二阶段方法, 则完成下一层子事务的二阶段方法调用)。

[0117] 从上述描述可知, 本发明实施例提供一种二阶段分布式事务对于路由服务节点的处理方法, 首先响应于分布式事务的一阶段调用及/或二阶段调用请求, 判断当前节点的事务场景类型是否为唯一的传递类型; 若是, 则根据事务场景对应的传递场景事务处理Bean, 以对事务场景进行一阶段处理; 接着, 传递处理后的事务场景至预先指定的下一层子事务节点; 最后下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。本发明通过Spring上下文的条件加载机制, 新增事务场景TCC\_TRANSMIT(传递类型), 在启动时, 数据库操作相关的bean(java类)不进行加载; 通过分布式服务Dubbo协议的Filter机制, 将路由服务节点和业务服务节点作为一个整体进行处理, 路由服务节点无需存储下一层子事务的事务信息, 通过二阶段操作显式的调用业务服务节点, 完成二阶段操作的调用。本发明消除路由服务节点的数据源, 并且支持路由服务节点的负载均衡机制, 具体地, 具有益效果如下:

[0118] 1. 解耦数据源: 基本Spring上下文的条件加载机制, 对于传递类型的事务场景, 屏蔽数据库操作相关的bean。

[0119] 2. 支持路由服务节点的负载均衡能力, 通过RpcContext的Filter机制, 设置二阶段操作标识, 显式的调用下一层子事务的二阶段操作。

[0120] 基于同一发明构思, 本申请实施例还提供了一种二阶段分布式事务对于路由服务节点的处理装置, 可以用于实现上述实施例所描述的方法, 如下面的实施例。由于二阶段分布式事务对于路由服务节点的处理装置解决问题的原理与二阶段分布式事务对于路由服务节点的处理方法相似, 因此二阶段分布式事务对于路由服务节点的处理装置的实施可以参见二阶段分布式事务对于路由服务节点的处理方法实施, 重复之处不再赘述。以下所使用的, 术语“单元”或者“模块”可以实现预定功能的软件和/或硬件的组合。尽管以下实施例所描述的系统较佳地以软件来实现, 但是硬件, 或者软件和硬件的组合的实现也是可能并被构想的。

[0121] 本发明的实施例提供一种能够实现二阶段分布式事务对于路由服务节点的处理方法的二阶段分布式事务对于路由服务节点的处理装置的具体实施方式, 参见图8, 二阶段分布式事务对于路由服务节点的处理装置具体包括如下内容:

[0122] 类型判断模块10, 用于响应于分布式事务的一阶段调用及/或二阶段调用请求, 判断当前节点的事务场景类型是否为唯一的传递类型;

[0123] 一阶段处理模块20, 用于若当前节点的事务场景的类型是唯一的传递类型, 则根

据所述事务场景对应的传递场景事务处理Bean,以对所述事务场景进行一阶段处理;

[0124] 事务场景传递模块30,用于传递处理后的事务场景至预先指定的下一层子事务节点;

[0125] 二阶段处理模块40,用于根据所述下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。

[0126] 一实施例中,所述类型判断模块具体用于根据配置值确定所述当前节点的事务场景类型。

[0127] 一实施例中,参见图9,所述一阶段处理模块20包括:

[0128] 一阶段ID确定单元201,用于利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;

[0129] 第一上下文写入单元202,用于将所述主事务ID以及所述子事务ID写入事务链路的第一上下文中;

[0130] 一阶段判断单元203,用于判断当前节点的事务场景是否为传递类型;

[0131] 一阶段第一处理单元204,用于若当前节点的事务场景为传递类型,则根据所述第一上下文以及当前节点在一阶段处理中指定的下一层子事务节点所对应的一阶段方法处理所述事务场景;

[0132] 一阶段第二处理单元205,用于若当前节点的事务场景不为传递类型,则根据事务数据库以及当前节点一阶段方法中制定的下一层子事务节点对应的一阶段方法处理所述事务场景。

[0133] 一实施例中,参见图10,所述二阶段处理模块40包括:

[0134] 二阶段ID确定单元401,用于利用分布式服务的Filter处理机制,获取所述事务场景的主事务ID和子事务ID;

[0135] 第二上下文写入单元402,用于将所述主事务ID和子事务ID写入事务链路的第二上下文中;

[0136] 二阶段判断单元403,用于判断当前节点的事务场景是否为传递类型;

[0137] 二阶段第一处理单元404,用于若当前节点的事务场景为传递类型,则根据所述第二上下文以及当前节点在二阶段处理中指定的下一层子事务节点所对应的二阶段方法处理所述事务场景;

[0138] 二阶段第二处理单元405,用于若当前节点的事务场景不为传递类型,则查询所述事务数据库,以获取所述当前节点的下一层子事务ID以及对应的方法,并对所述事务场景进行处理。

[0139] 从上述描述可知,本发明实施例提供一种二阶段分布式事务对于路由服务节点的处理方法及装置,首先响应于分布式事务的一阶段调用及/或二阶段调用请求,判断当前节点的事务场景类型是否为唯一的传递类型;若是,则根据事务场景对应的传递场景事务处理Bean,以对事务场景进行一阶段处理;接着,传递处理后的事务场景至预先指定的下一层子事务节点;最后下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。具体地,具有益效果如下:

[0140] 1.解耦数据源:基本Spring上下文的条件加载机制,对于传递类型的事务场景,屏蔽数据库操作相关的bean。

[0141] 2.支持路由服务节点的负载均衡能力,通过RpcContext的Filter机制,设置二阶段操作标识,显式的调用下一层子事务的二阶段操作。

[0142] 本申请的实施例还提供能够实现上述实施例中的二阶段分布式事务对于路由服务节点的处理方法中全部步骤的一种电子设备的具体实施方式,参见图11,电子设备具体包括如下内容:

[0143] 处理器(processor) 1201、存储器(memory) 1202、通信接口(Communications Interface) 1203和总线1204;

[0144] 其中,处理器1201、存储器1202、通信接口1203通过总线1204完成相互间的通信;通信接口1203用于实现服务器端设备以及客户端设备等相关设备之间的信息传输。

[0145] 处理器1201用于调用存储器1202中的计算机程序,处理器执行计算机程序时实现上述实施例中的二阶段分布式事务对于路由服务节点的处理方法中的全部步骤,例如,处理器执行计算机程序时实现下述步骤:

[0146] 步骤100:响应于分布式事务的一阶段调用及/或二阶段调用请求,判断当前节点的事务场景类型是否为唯一的传递类型;

[0147] 步骤200:若当前节点的事务场景的类型是唯一的传递类型,则根据所述事务场景对应的传递场景事务处理Bean,以对所述事务场景进行一阶段处理;

[0148] 步骤300:传递处理后的事务场景至预先指定的下一层子事务节点;

[0149] 步骤400:根据所述下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。

[0150] 本申请的实施例还提供能够实现上述实施例中的二阶段分布式事务对于路由服务节点的处理方法中全部步骤的一种计算机可读存储介质,计算机可读存储介质上存储有计算机程序,该计算机程序被处理器执行时实现上述实施例中的二阶段分布式事务对于路由服务节点的处理方法的全部步骤,例如,处理器执行计算机程序时实现下述步骤:

[0151] 步骤100:响应于分布式事务的一阶段调用及/或二阶段调用请求,判断当前节点的事务场景类型是否为唯一的传递类型;

[0152] 步骤200:若当前节点的事务场景的类型是唯一的传递类型,则根据所述事务场景对应的传递场景事务处理Bean,以对所述事务场景进行一阶段处理;

[0153] 步骤300:传递处理后的事务场景至预先指定的下一层子事务节点;

[0154] 步骤400:根据所述下一层子事务节点对处理后的事务场景进行分布式事务二阶段处理。

[0155] 本说明书中的各个实施例均采用递进的方式描述,各个实施例之间相同相似的部分互相参见即可,每个实施例重点说明的都是与其他实施例的不同之处。尤其,对于硬件+程序类实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[0156] 上述对本说明书特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下,在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外,在附图中描绘的过程不一定要示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中,多任务处理和并行处理也是可以的或者可能是有利的。

[0157] 虽然本申请提供了如实施例或流程图的方法操作步骤,但基于常规或者无创造性的劳动可以包括更多或者更少的操作步骤。实施例中列举的步骤顺序仅仅为众多步骤执行顺序中的一种方式,不代表唯一的执行顺序。在实际中的装置或客户端产品执行时,可以按照实施例或者附图所示的方法顺序执行或者并行执行(例如并行处理器或者多线程处理的环境)。

[0158] 为了描述的方便,描述以上装置时以功能分为各种模块分别描述。当然,在实施本说明书实施例时可以把各模块的功能在同一个或多个软件和/或硬件中实现,也可以将实现同一功能的模块由多个子模块或子单元的组合实现等。以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,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。

[0159] 本领域技术人员也知道,除了以纯计算机可读程序代码方式实现控制器以外,完全可以通过将方法步骤进行逻辑编程来使得控制器以逻辑门、开关、专用集成电路、可编程逻辑控制器和嵌入微控制器等的形式来实现相同功能。因此这种控制器可以被认为是一种硬件部件,而对其内部包括的用于实现各种功能的装置也可以视为硬件部件内的结构。或者甚至,可以将用于实现各种功能的装置视为既可以是实现方法的软件模块又可以是硬件部件内的结构。

[0160] 在一个典型的配置中,计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[0161] 内存可能包括计算机可读介质中的非永久性存储器,随机存取存储器(RAM)和/或非易失性内存等形式,如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

[0162] 本说明书实施例可以在由计算机执行的计算机可执行指令的一般上下文中描述,例如程序模块。一般地,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本说明书实施例,在这些分布式计算环境中,由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0163] 本说明书中的各个实施例均采用递进的方式描述,各个实施例之间相同相似的部分互相参见即可,每个实施例重点说明的都是与其他实施例的不同之处。尤其,对于系统实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。在本说明书的描述中,参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包含于本说明书实施例的至少一个实施例或示例中。在本说明书中,对上述术语的示意性表述不必须针对的是相同的实施例或示例。而且,描述的具体特征、结构、材料或者特点可以在任一个或多个实施例或示例中以合适的方式结合。此外,在不相互矛盾的情况下,本领域的技术人员可以将本说明书中描述的不同实施例或示例以及不同实施例或示例的特征进行结合和组合。

[0164] 以上所述仅为本说明书实施例的实施例而已,并不用于限制本说明书实施例。对

于本领域技术人员来说,本说明书实施例可以有各种更改和变化。凡在本说明书实施例的精神和原理之内所作的任何修改、等同替换、改进等,均应包含在本说明书实施例的权利要求范围之内。

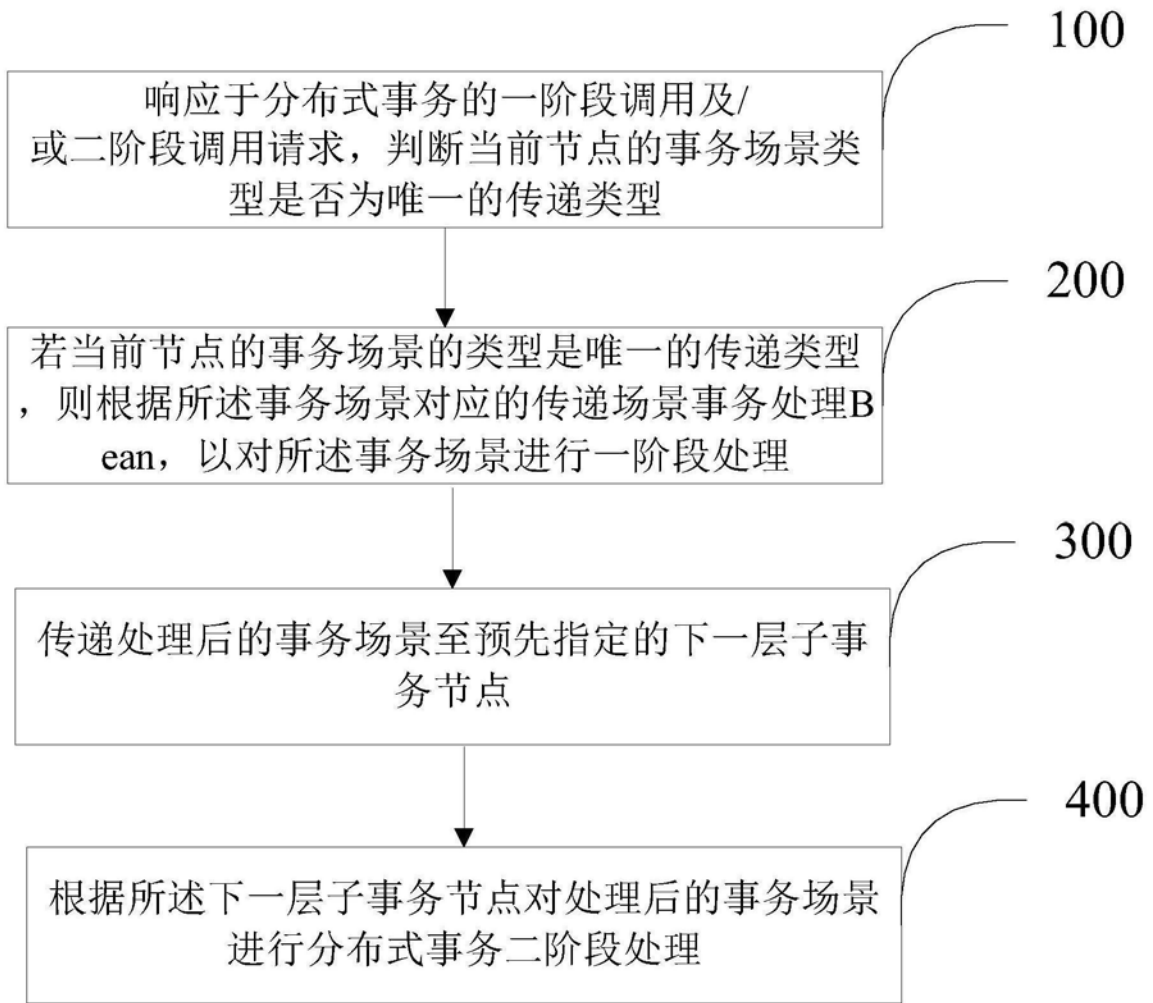


图1



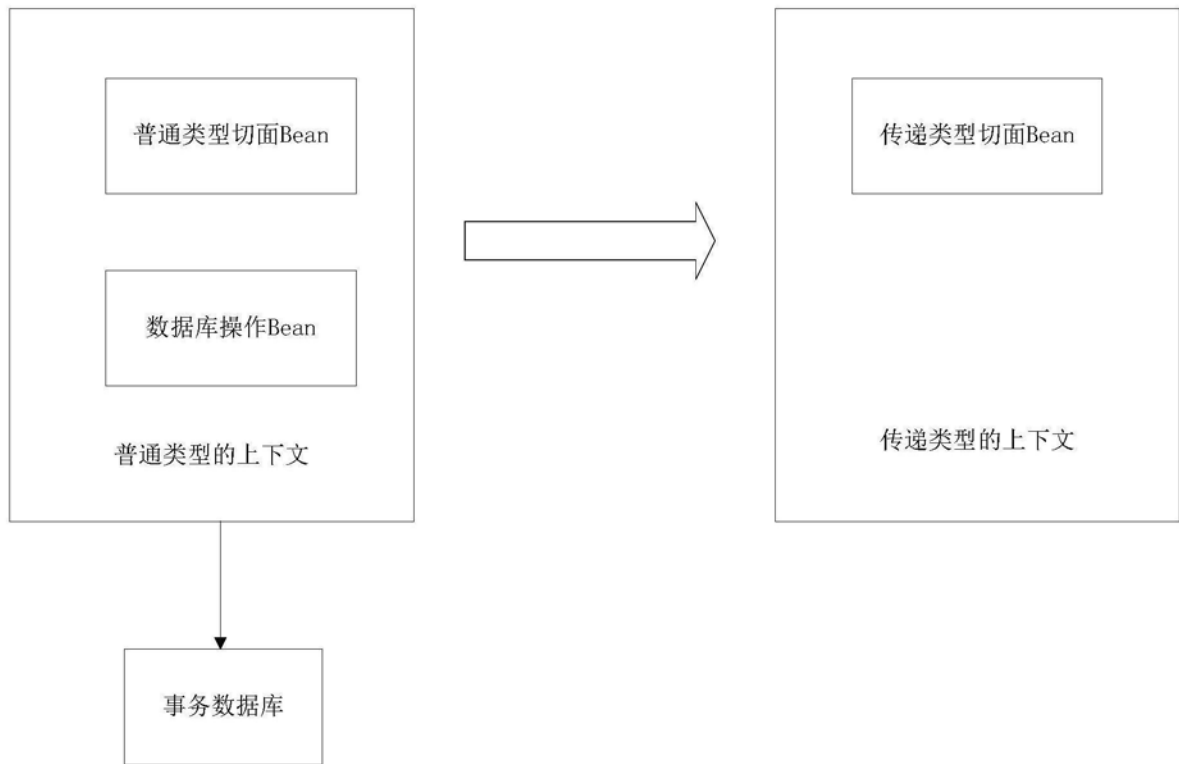


图2

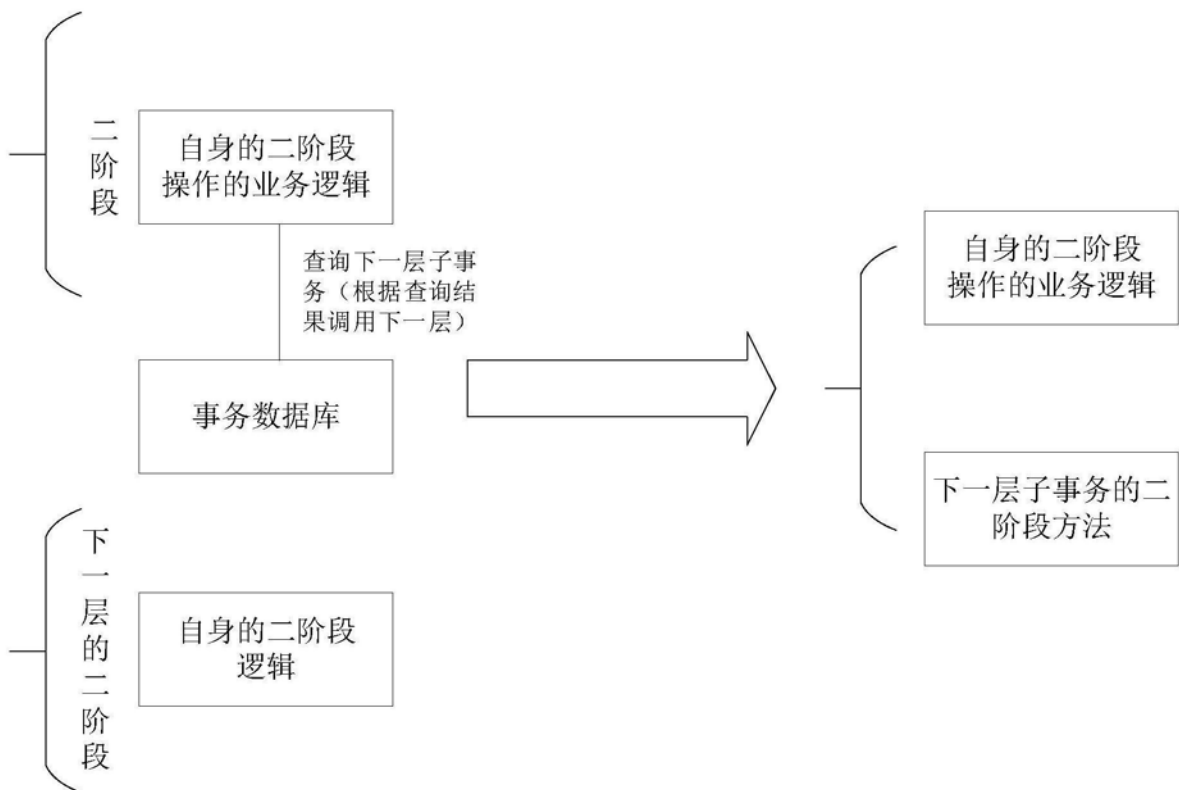


图3

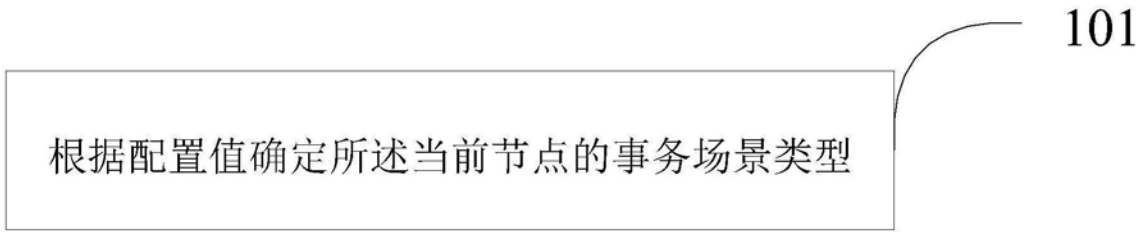


图4

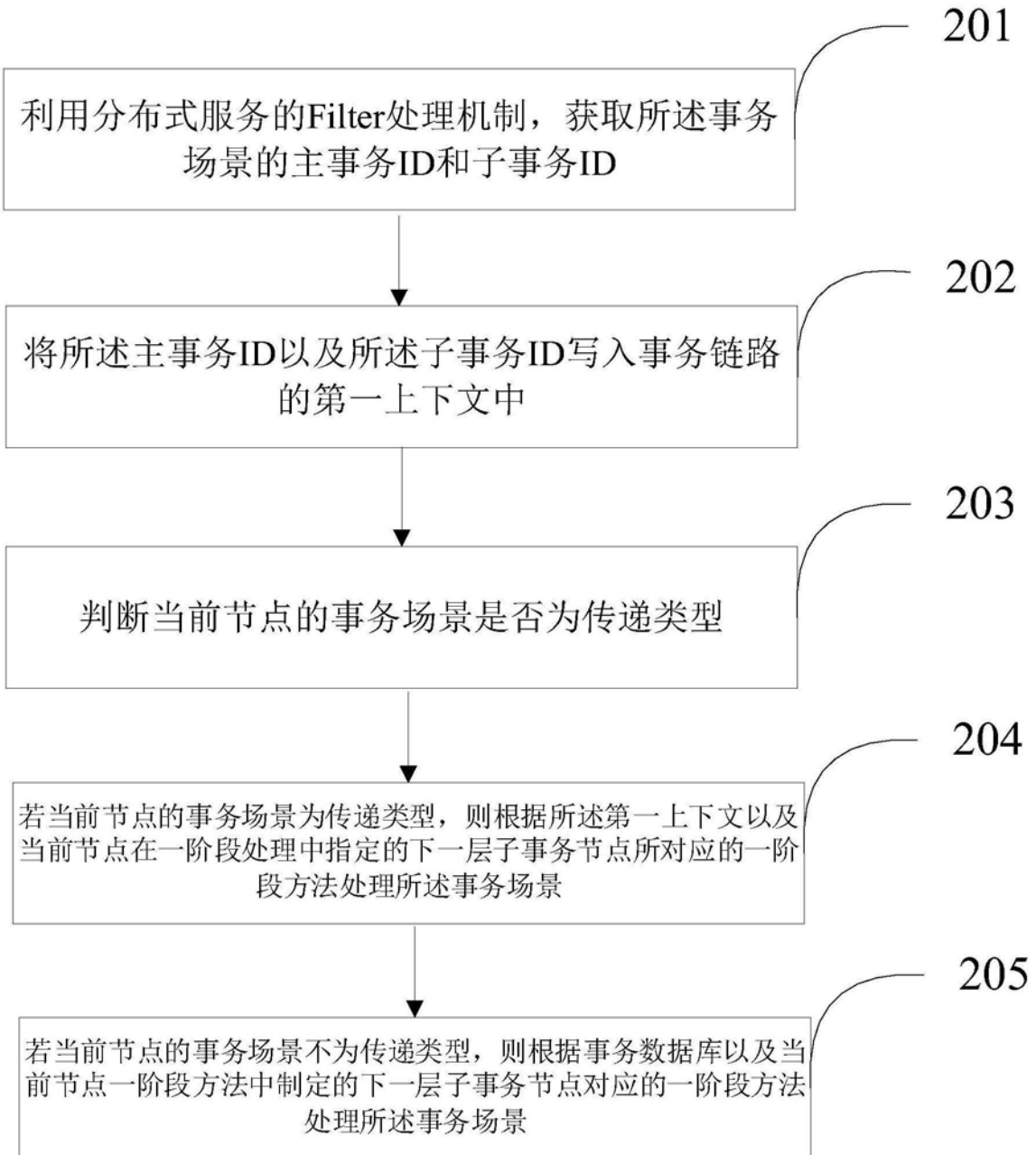


图5

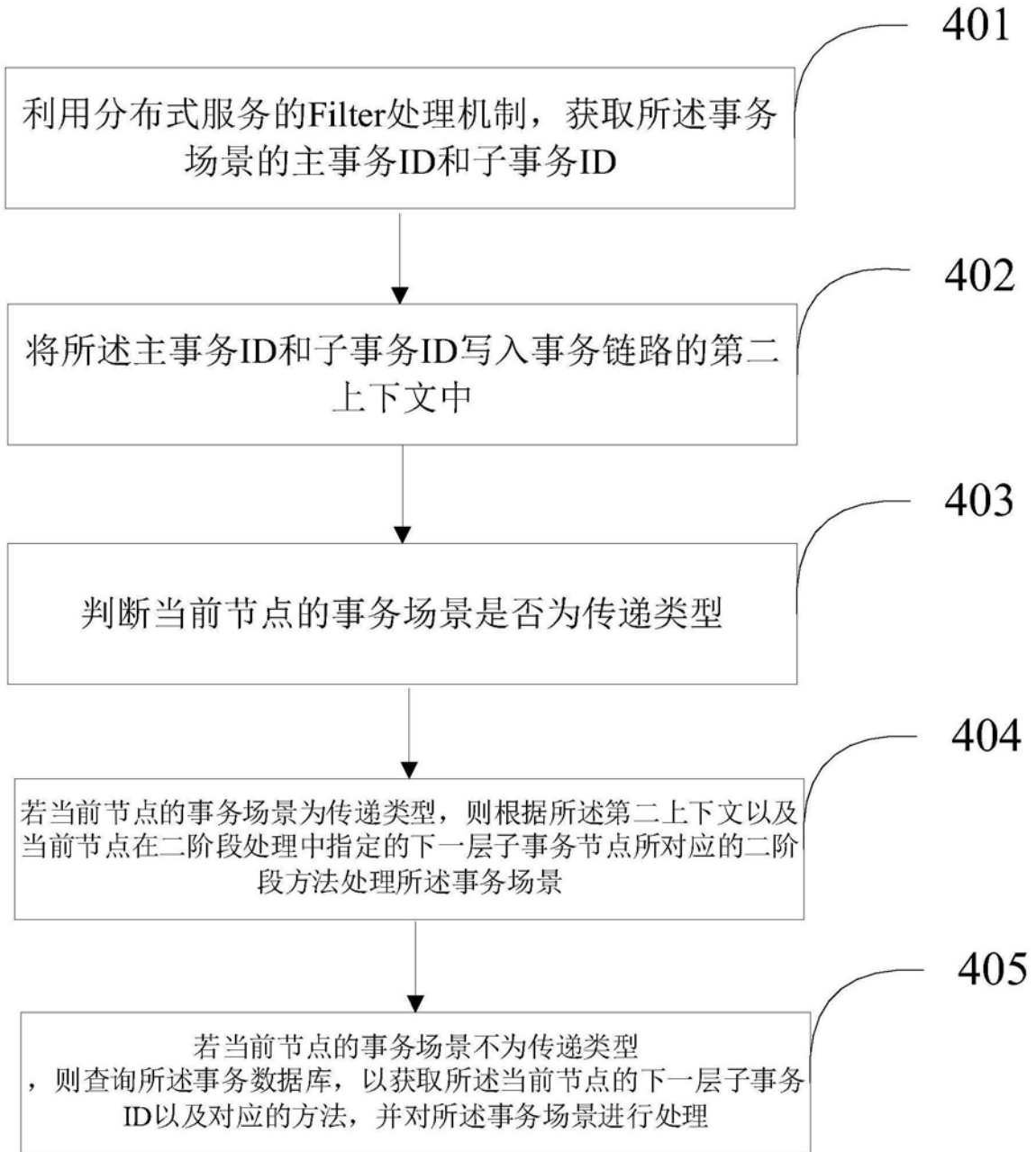


图6

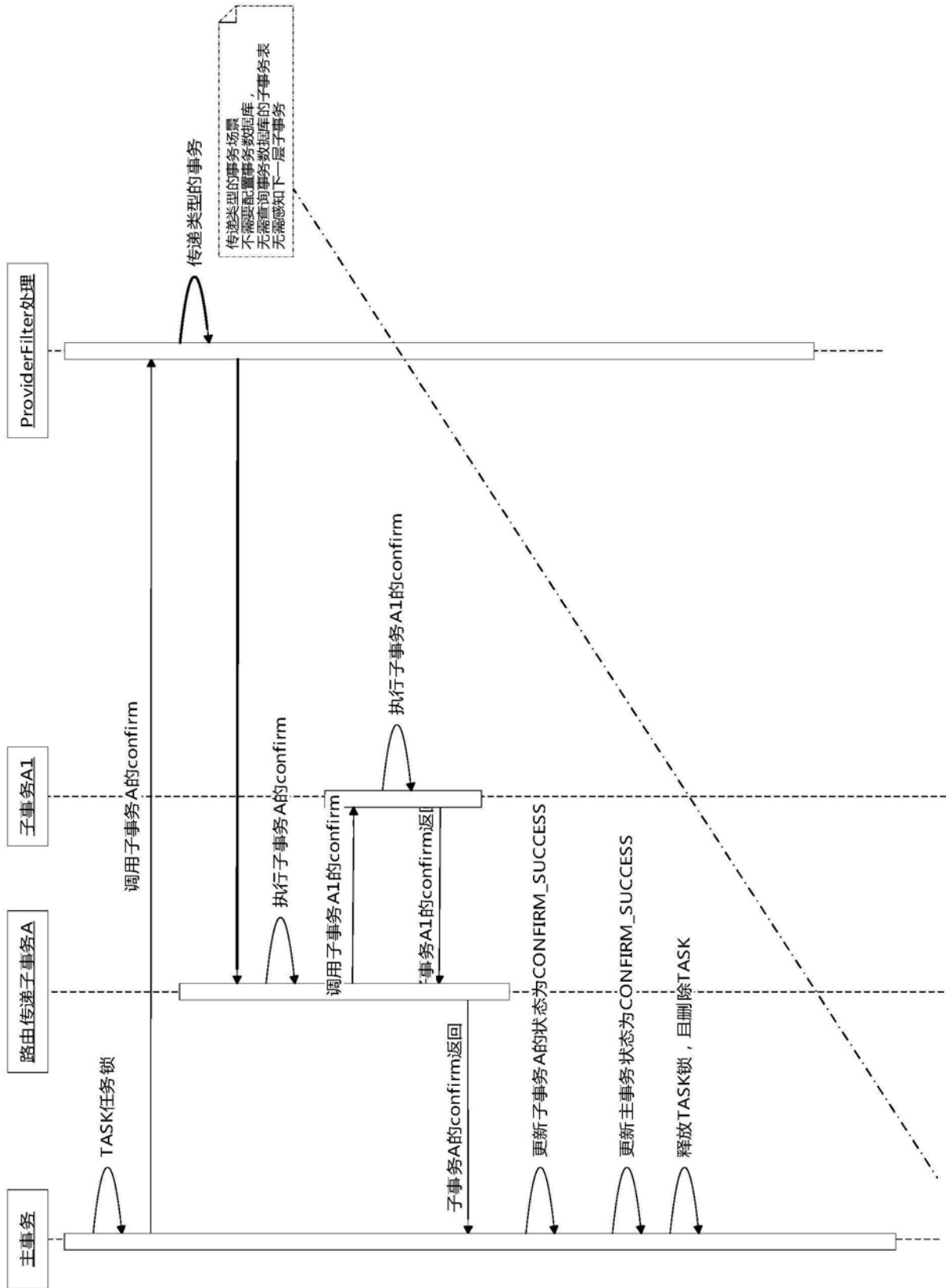


图7

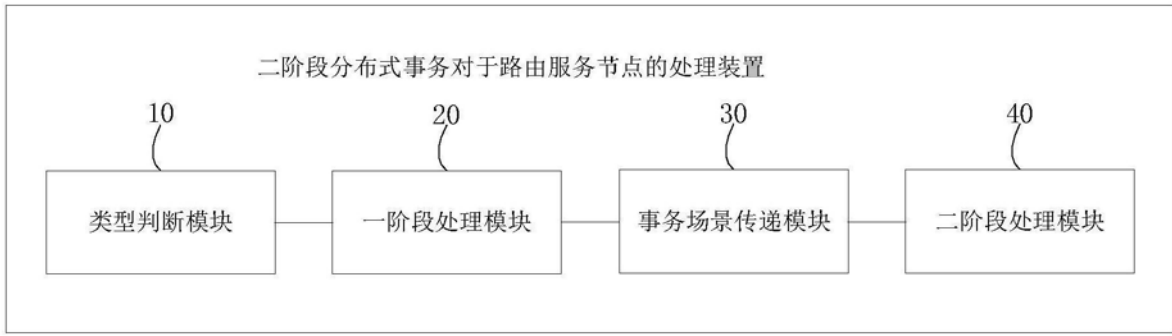


图8

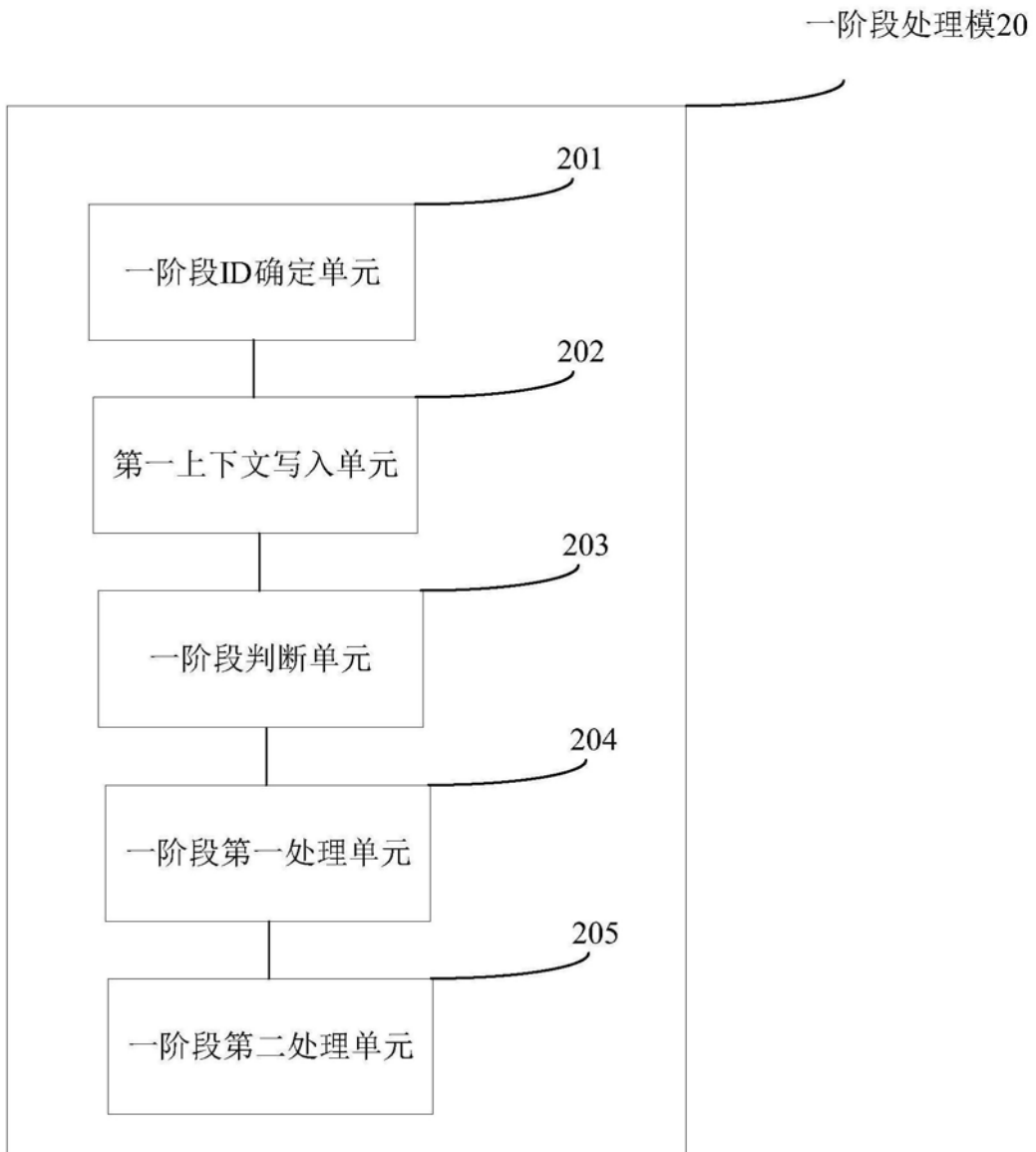


图9

二阶段处理模块40

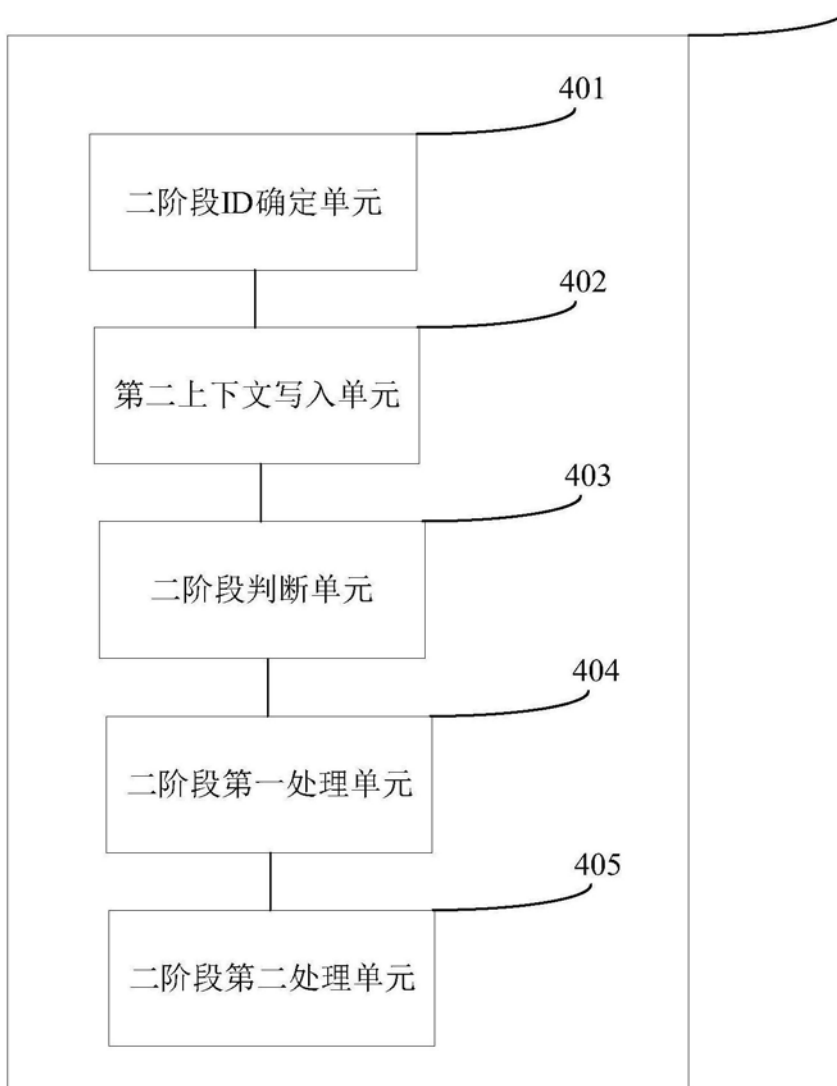


图10

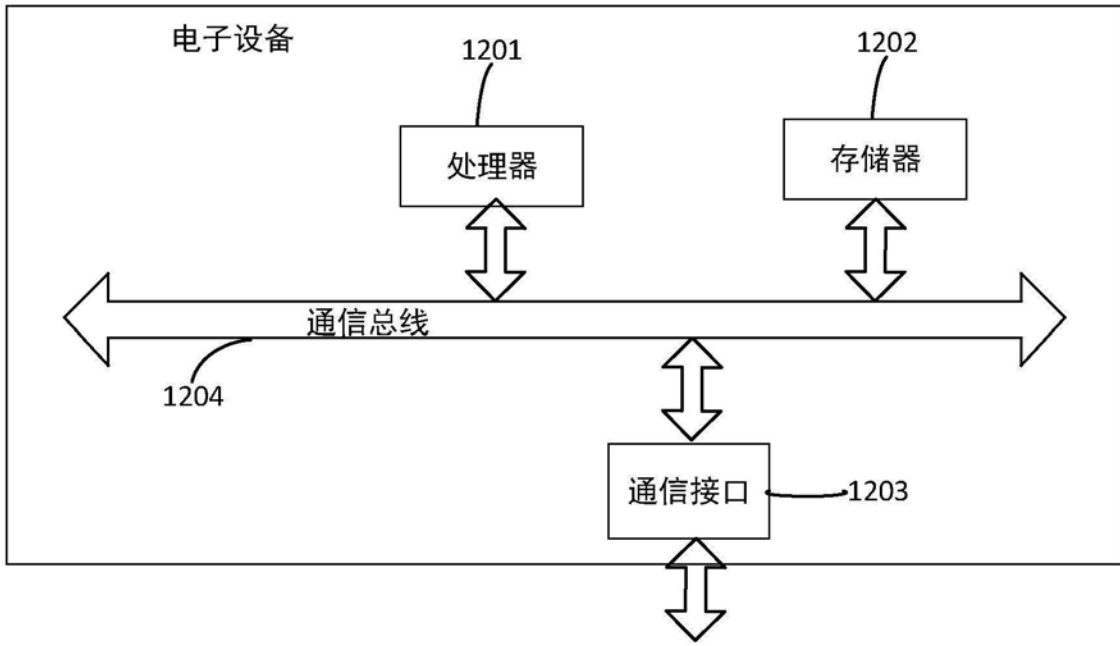


图11