



# (12) 发明专利申请

(10) 申请公布号 CN 116302702 A

(43) 申请公布日 2023. 06. 23

(21) 申请号 202310273295.9

(22) 申请日 2023.03.20

(71) 申请人 阿里云计算有限公司

地址 310000 浙江省杭州市西湖区转塘科技经济区块12号

(72) 发明人 申海强

(74) 专利代理机构 广州铸智知识产权代理有限公司 44886

专利代理师 郗名悦 徐瑞红

(51) Int. Cl.

G06F 11/14 (2006.01)

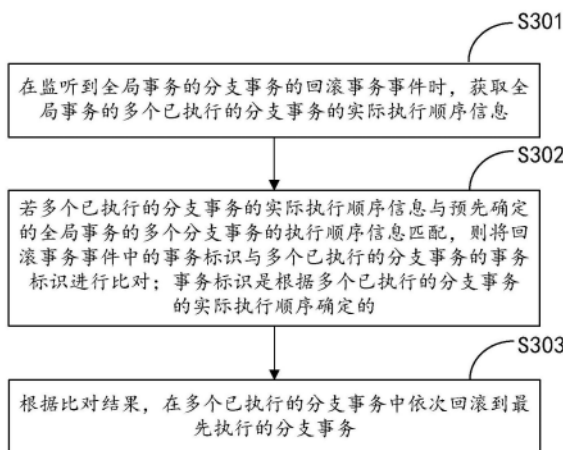
权利要求书2页 说明书9页 附图3页

## (54) 发明名称

分布式事务回滚方法、电子设备及存储介质

## (57) 摘要

本申请提供了一种分布式事务回滚方法、电子设备及存储介质,涉及云计算技术领域,该方法包括:在监听到全局事务的分支事务的回滚事务事件时,获取全局事务的多个已执行的分支事务的实际执行顺序信息;若多个已执行的分支事务的实际执行顺序信息与预先确定的全局事务的多个分支事务的执行顺序信息匹配,则将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比对;根据比对结果,在多个已执行的分支事务中依次回滚到最先执行的分支事务。本实施例中,对于实际执行顺序与预先确定的执行顺序相匹配的分支事务,则采用基于回滚事务事件驱动的方式,不需要引入流程引擎,可以降低事务回滚的实现成本。



1. 一种分布式事务回滚方法,其特征在于,所述方法包括:

在监听到全局事务的分支事务的回滚事务事件时,获取所述全局事务的多个已执行的分支事务的实际执行顺序信息;

若所述多个已执行的分支事务的实际执行顺序信息与预先确定的所述全局事务的多个分支事务的执行顺序信息匹配,则将所述回滚事务事件中的事务标识与所述多个已执行的分支事务的事务标识进行比对;所述事务标识是根据所述多个已执行的分支事务的实际执行顺序确定的;

根据比对结果,在所述多个已执行的分支事务中依次回滚到最先执行的分支事务。

2. 根据权利要求1所述的方法,其特征在于,所述事务标识为事务顺序编号,所述将所述回滚事务事件中的事务标识与所述多个已执行的分支事务的事务标识进行比对,包括:

确定所述回滚事务事件中的事务顺序编号与所述多个已执行的分支事务的事务顺序编号之间的差值。

3. 根据权利要求2所述的方法,其特征在于,所述根据比对结果,在所述多个已执行的分支事务中依次回滚到最先执行的分支事务,包括:

若所述回滚事务事件中的事务顺序编号与所述已执行的分支事务的事务顺序编号之间的差值为预设值,则将所述已执行的分支事务确定为目标分支事务;

从多个所述目标分支事务依次回滚到所述多个已执行的分支事务中最先执行的分支事务。

4. 根据权利要求3所述的方法,其特征在于,所述从多个所述目标分支事务依次回滚到所述多个已执行的分支事务中最先执行的分支事务,包括:

回滚到所述目标分支事务,生成所述目标分支事务的回滚事务事件;

根据当前的所述目标分支事务的回滚事务事件,回滚到下一目标分支事务;

从多个所述目标分支事务,依次回滚到所述多个已执行的分支事务中最先执行的分支事务。

5. 根据权利要求4所述的方法,其特征在于,所述根据当前的所述目标分支事务的回滚事务事件,回滚到下一目标分支事务,包括:

根据当前的所述目标分支事务的回滚事务事件中的事务顺序编号,确定下一目标分支事务;

根据当前的所述目标分支事务的回滚事务事件中的上下文信息,回滚到所述下一目标分支事务。

6. 根据权利要求4所述的方法,其特征在于,所述从多个所述目标分支事务,依次回滚到所述多个已执行的分支事务中最先执行的分支事务,还包括:

从当前的所述目标分支事务依次回滚到所述多个已执行的分支事务中最先执行的分支事务,并生成所述最先执行的分支事务的回滚事务事件。

7. 根据权利要求1-6任一项所述的方法,其特征在于,所述获取所述全局事务的多个已执行的分支事务的实际执行顺序信息,包括:

获取所述全局事务的多个已执行的分支事务分别对应的顺序标记日志;所述分支事务对应的顺序标记日志为所述分支事务执行完成时生成的;

根据多个所述顺序标记日志,确定多个已执行的分支事务的实际执行顺序信息。

8. 根据权利要求1-6任一项所述的方法,其特征在于,所述方法还包括:

若所述多个已执行的分支事务的实际执行顺序信息与预先确定的所述全局事务的多个分支事务的执行顺序信息不匹配,则启动流程引擎,利用所述流程引擎驱动所述全局事务的分支事务进行事务回滚。

9. 根据权利要求1-6任一项所述的方法,其特征在于,所述分支事务包括微服务。

10. 一种电子设备,其特征在于,包括存储器、处理器及存储在存储器上的计算机程序,所述处理器在执行所述计算机程序时实现权利要求1-9中任一项所述的方法。

11. 一种计算机可读存储介质,所述计算机可读存储介质内存储有计算机程序,所述计算机程序被处理器执行时实现权利要求1-9中任一项所述的方法。

## 分布式事务回滚方法、电子设备及存储介质

### 技术领域

[0001] 本申请涉及云计算技术领域,尤其涉及一种分布式事务回滚方法、电子设备及存储介质。

### 背景技术

[0002] 微服务架构是一种用于构建应用的架构方案。微服务架构有别于更为传统的单体式方案,可将应用拆分成多个核心功能。每个功能都被称为一项服务,可以单独构建和部署,各项服务在工作 and 出现故障时不会相互影响。

[0003] 微服务架构下,分布式事务通常需要Saga模式来管理,Saga模式是一种在分布式事务场景中跨微服务管理数据一致性的方法。Saga模式下的事务回滚目前的解决方案是使用编排(Orchestration)的方式,采用流程引擎来驱动,事务链路较复杂,系统实现成本较高。

### 发明内容

[0004] 本申请实施例提供一种分布式事务回滚方法、电子设备及存储介质,以降低事务回滚的实现成本。

[0005] 第一方面,本申请实施例提供了一种分布式事务回滚方法,包括:

[0006] 在监听到全局事务的分支事务的回滚事务事件时,获取全局事务的多个已执行的分支事务的实际执行顺序信息;若多个已执行的分支事务的实际执行顺序信息与预先存储的全局事务的多个分支事务的执行顺序信息匹配,则将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比对;事务标识是根据多个已执行的分支事务的实际执行顺序确定的;

[0007] 根据比对结果,在多个已执行的分支事务中依次回滚到最先执行的分支事务。

[0008] 第二方面,本申请实施例提供了一种电子设备,包括存储器、处理器及存储在存储器上的计算机程序,处理器在执行计算机程序时实现上述任一项的方法。

[0009] 第三方面,本申请实施例提供了一种计算机可读存储介质,计算机可读存储介质内存储有计算机程序,计算机程序被处理器执行时实现上述任一项的方法。

[0010] 与现有技术相比,本申请具有如下优点:

[0011] 本申请提供了一种分布式事务回滚方法、电子设备及存储介质,在监听到全局事务的分支事务的回滚事务事件时,获取全局事务的多个已执行的分支事务的实际执行顺序信息;若多个已执行的分支事务的实际执行顺序信息与预先确定的全局事务的多个分支事务的执行顺序信息匹配,则将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比对;事务标识是根据多个已执行的分支事务的实际执行顺序确定的;根据比对结果,在多个已执行的分支事务中依次回滚到最先执行的分支事务。本实施例中,对于按照确定的执行顺序执行的分支事务,即该分支事务的实际执行顺序与预先确定的执行顺序相匹配,则通过将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比

对,实现已执行的分支事务的回滚,采用基于回滚事务事件驱动的方式,不需要引入流程引擎,可以降低事务回滚的实现成本。

[0012] 上述说明仅是本申请技术方案的概述,为了能够更清楚了解本申请的技术手段,可依照说明书的内容予以实施,并且为了让本申请的上述和其他目的、特征和优点能够更明显易懂,以下特举本申请的具体实施方式。

### 附图说明

[0013] 在附图中,除非另外规定,否则贯穿多个附图相同的附图标记表示相同或相似的部件或元素。这些附图不一定是按照比例绘制的。应该理解,这些附图仅描绘了根据本申请的一些实施方式,而不应将其视为是对本申请范围的限制。

[0014] 图1为本申请提供的分布式事务回滚方法的一个应用场景示意图;

[0015] 图2为本申请一实施例的分布式事务回滚过程的示意图;

[0016] 图3是本申请一实施例的分布式事务回滚方法的流程图;

[0017] 图4是本申请一实施例的数据安全检测装置的结构框图;以及

[0018] 图5为用来实现本申请实施例的电子设备的框图。

### 具体实施方式

[0019] 在下文中,仅简单地描述了某些示例性实施例。正如本领域技术人员可认识到的那样,在不脱离本申请的构思或范围的情况下,可通过各种不同方式修改所描述的实施例。因此,附图和描述被认为本质上是示例性的,而非限制性的。

[0020] 为便于理解本申请实施例的技术方案,以下对本申请实施例的相关技术进行说明。以下相关技术作为可选方案与本申请实施例的技术方案可以进行任意结合,其均属于本申请实施例的保护范围。

[0021] 图1为本申请提供的分布式事务回滚方法的一个应用场景示意图。本申请实施例中的方案可以应用于微服务架构中,全局事务为分布式事务,分支事务为微服务。如图1所示,全局事务Global Tx包括多个分支事务(本地事务):Local Tx1、Local Tx2、Local Tx3、Local Tx4、Local Tx5和Local Tx6。预先确定该全局事务的各分支事务的执行顺序,得到执行顺序信息。例如,预先确定各分支事务各自对应的事务顺序编号,Local Tx1、Local Tx2、Local Tx3、Local Tx4、Local Tx5和Local Tx6对应的事务顺序编号分别为1、2、3、4、5、6。

[0022] 在事务实际执行过程中,分支事务有可能是按照预先确定的执行顺序执行的,也有可能不是按照预先确定的执行顺序执行的,实际的执行顺序还要根据执行完成之后的结果来确定。每个分支事务执行完成后,生成该分支事务对应的顺序标记日志,日志中记载事务的标识。根据多个已执行分支事务分别对应的顺序标记日志,可以得到多个已执行分支事务的实际执行顺序。

[0023] 图2为本申请提供的分布式事务回滚过程的示意图。如图2所示,若监听到全局事务的分支事务Local Tx4的回滚事务事件(Rollback Event,RBEvent),如图中所示的RBEvent4,RBEvent4中包括Local Tx4的事务顺序编号4,获取预先确定的该全局事务的分支事务Local Tx1、Local Tx2、Local Tx3、Local Tx4的执行顺序信息,与根据Local Tx1、

Local Tx2、Local Tx3、Local Tx4的顺序标记日志确定的实际执行顺序信息进行比对,如果匹配,则将事务顺序编号4与多个已执行的分支事务Local Tx1、Local Tx2、Local Tx3的事务顺序编号进行比较,RBEvent4中的事务顺序编号4比Local Tx3的事务顺序号3大1,则回滚到Local Tx3,并生成Local Tx3的RBEvent3,再次进行事务顺序编号比较,回滚到Local Tx2,并生成Local Tx2的RBEvent2,按照之前的方式继续回滚,直到回滚到Local Tx1,生成RBEvent1,结束回滚(如图中所示的Done)。

[0024] 相关技术中,无论是否是按照确定的执行顺序执行的分支事务,都是利用流程引擎来整体控制事务回滚顺序,即用编排(Orchestration)的方式实现事务回滚。这种处理方式,需要记录流程实例的运行时信息,需要较大的系统开销。

[0025] 本实施例中,对于按照确定的执行顺序执行的分支事务,即该分支事务的实际执行顺序与预先确定的执行顺序相匹配,通过将回滚事务事件中的事务顺序编号与多个已执行的分支事务的事务顺序编号进行比对,实现已执行的分支事务的回滚,仅使用协调(Choreography)方案的事件驱动,即保证了按顺序回滚,不引入重量级的编排(Orchestration)方案所依赖的流程引擎,可以降低事务回滚的实现成本。

[0026] 本申请实施例提供了一种分布式事务回滚方法,本实施例中的方法可以应用计算设备中,计算设备可以包括:服务器、用户终端等。如图3所示为本申请一实施例的分布式事务回滚方法的流程图,该方法包括:

[0027] 步骤S301,在监听到全局事务的分支事务的回滚事务事件时,获取全局事务的多个已执行的分支事务的实际执行顺序信息。

[0028] 步骤S302,若多个已执行的分支事务的实际执行顺序信息与预先确定的全局事务的多个分支事务的执行顺序信息匹配,则将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比对;事务标识是根据多个已执行的分支事务的实际执行顺序确定的。

[0029] 步骤S303,根据比对结果,在多个已执行的分支事务中依次回滚到最先执行的分支事务。

[0030] 其中,事务由事务开始和事务结束之间执行的操作组成。事务具有的属性包括:原子性、一致性、隔离性和持续性。全局事务为分布式事务,分支事务为本地事务。本地事务在单个数据源上进行数据的访问和更新,分布式事务跨越多个数据源来进行数据的访问和更新。

[0031] 预先确定全局事务的各分支事务的执行顺序,得到执行顺序信息。其中,执行顺序信息可以根据具体需要或者相应的规则进行配置,又或者根据之前多个用户的操作流程的顺序来确定,本申请对此不作限定。在事务实际执行过程中,分支事务有可能是按照预先确定的执行顺序执行的,也有可能不是按照预先确定的执行顺序执行的,实际的执行顺序还要根据执行完成之后的结果来确定。

[0032] 在监听到回滚事务事件时,通过比对多个已执行的分支事务的实际执行顺序信息与预先确定的全局事务的多个分支事务的执行顺序信息是否匹配,来确定具体的事务回滚策略。

[0033] 若多个已执行的分支事务的实际执行顺序信息与预先确定的全局事务的多个分支事务的执行顺序信息匹配,则不需要启动流程引擎,仅需要通过回滚事务事件触发分支

事务的回滚。根据多个已执行的分支事务的实际执行顺序确定事务标识,将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比对,根据比对结果可以确定出回滚事务事件回滚到哪一个分支事务,再依次按顺序回滚到多个已执行的分支事务中最先执行的分支事务。

[0034] 示例性的,分支事务包括微服务。微服务是指将单一应用程序划分成的小的服务,微服务之间互相协调、互相配合。每个微服务运行在独立的进程中,微服务与微服务间采用轻量级的通信机制互相沟通。

[0035] 本申请提供了一种分布式事务回滚方法,在监听到全局事务的分支事务的回滚事务事件时,获取全局事务的多个已执行的分支事务的实际执行顺序信息;若多个已执行的分支事务的实际执行顺序信息与预先确定的全局事务的多个分支事务的执行顺序信息匹配,则将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比对;事务标识是根据多个已执行的分支事务的实际执行顺序确定的;根据比对结果,在多个已执行的分支事务中依次回滚到最先执行的分支事务。本实施例中,对于按照确定的执行顺序执行的分支事务,即该分支事务的实际执行顺序与预先确定的执行顺序相匹配,则通过将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比对,实现已执行的分支事务的回滚,采用基于回滚事务事件驱动的方式,不需要引入流程引擎,可以降低事务回滚的实现成本。

[0036] 其中,步骤S301的具体实现方式见如下实施例:

[0037] 在一种实现方式中,事务标识为事务顺序编号,将回滚事务事件中的事务标识与全局事务的多个已执行的分支事务的事务标识进行比对,包括:确定回滚事务事件中的事务顺序编号与多个已执行的分支事务的事务顺序编号之间的差值。

[0038] 在实际应用中,事务标识可以为事务顺序编号,事务顺序编号可以是阿拉伯数字编号,将回滚事务事件中的事务标识与全局事务的多个已执行的分支事务的事务标识进行比对的具体实现方式,可以包括:对于每个已执行的分支事务,将回滚事务事件中的事务顺序编号与已执行的分支事务的事务顺序编号计算差值,得到多个差值,根据多个差值来确定回滚到哪一个分支事务。

[0039] 在一种实现方式中,步骤S302,根据比对结果,在多个已执行的分支事务中依次回滚到最先执行的分支事务,包括:若回滚事务事件中的事务顺序编号与已执行的分支事务的事务顺序编号之间的差值为预设值,则将已执行的分支事务确定为目标分支事务;从多个目标分支事务依次回滚到多个已执行的分支事务中最先执行的分支事务。

[0040] 在实际应用中,在事务标识为事务顺序编号的情况下,对于回滚事务事件中的事务顺序编号与多个已执行的分支事务的事务顺序编号之间的差值,如果其中一个差值为预设值,例如,预设值可以为1,则将该已执行的事务顺序编号对应的分支事务作为目标分支事务,回滚到目标分支事务,再从目标分支事务依次回滚到多个已执行的分支事务中最先执行的分支事务。

[0041] 在一种实现方式中,从多个目标分支事务依次回滚到多个已执行的分支事务中最先执行的分支事务,包括:回滚到目标分支事务,生成目标分支事务的回滚事务事件;根据当前的目标分支事务的回滚事务事件,回滚到下一目标分支事务;从多个目标分支事务,依次回滚到多个已执行的分支事务中最先执行的分支事务。

[0042] 其中,回滚到目标分支事务之后,生成目标分支事务的回滚事务事件,目标分支事务的回滚事务事件中包含目标分支事务的事务顺序编号,继续将目标分支事务的事务顺序编号与已执行的分支事务的事务顺序编号进行比较,计算目标分支事务的事务顺序编号与多个已执行的分支事务的事务顺序编号之间的差值,如果其中一个差值为预设值,则得到下一目标分支事务,依此方式继续依次进行回滚,直到回滚到多个已执行的分支事务中最先执行的分支事务。

[0043] 在一种实现方式中,根据当前的目标分支事务的回滚事务事件,回滚到下一目标分支事务,包括:根据当前的目标分支事务的回滚事务事件中的事务顺序编号,确定下一目标分支事务;根据当前的目标分支事务的回滚事务事件中的上下文信息,回滚到下一目标分支事务。

[0044] 其中,回滚事务事件中包含事务顺序编号和上下文信息,上下文信息包括应用执行流程链路信息,包括应用执行过程中调用了哪些接口函数等。根据回滚事务事件中的事务顺序编号与已执行的分支事务的事务顺序编号的差值,确定下一目标分支事务是哪一个已执行的分支事务,利用上下文信息,回滚到下一目标分支事务。

[0045] 在一种实现方式中,从多个目标分支事务,依次回滚到多个已执行的分支事务中最先执行的分支事务,还包括:从当前的目标分支事务依次回滚到多个已执行的分支事务中最先执行的分支事务,并生成最先执行的分支事务的回滚事务事件。

[0046] 在结合图1和图2所示的示例中,当Local Tx4执行失败时,监测到生成Local Tx4的RBEvent4,从Local Tx4回滚到Local Tx3,生成Local Tx3的RBEvent3,从Local Tx3回滚到Local Tx2,生成Local Tx2的RBEvent2,从Local Tx2回滚到Local Tx1,Local Tx1最先执行的分支事务,生成Local Tx1的Local Tx1,从而完成整个回滚过程。

[0047] 在一具体的示例中,全局事务可以是用于网络购物的应用程序,其中可以包括多个微服务:订单服务、消费者服务、库存增减服务和支付服务。其中,订单服务用于创建订单;消费者服务用于确定当前订单的消费者是否有权限下单;库存增减服务用于对订单中的商品的库存数量进行增减;支付服务用于消费者实现订单支付。若支付服务发现订单支付失败,则回滚到库存增减服务,库存增减服务将扣减的库存再增加回去,依此类推,最终回滚到订单服务,撤销创建的订单。

[0048] 在一种实现方式中,步骤S301中,获取全局事务的多个已执行的分支事务的实际执行顺序信息,包括:获取全局事务的多个已执行的分支事务分别对应的顺序标记日志;分支事务对应的顺序标记日志为分支事务执行完成时生成的;根据多个顺序标记日志,确定多个已执行的分支事务的实际执行顺序信息。

[0049] 在实际应用中,每个分支事务执行完成后,生成该分支事务对应的顺序标记日志,日志中记载事务的标识。根据多个已执行分支事务分别对应的顺序标记日志,可以得到多个已执行分支事务的实际执行顺序。

[0050] 在一种实现方式中,该方法还包括:若多个已执行的分支事务的实际执行顺序信息与预先确定的全局事务的多个分支事务的执行顺序信息不匹配,则启动流程引擎,利用流程引擎驱动全局事务的分支事务进行事务回滚。

[0051] 在实际应用中,对于不是按照确定的执行顺序执行的全局事务的分支事务,通过启动流程引擎的方式进行事务回滚。



[0052] 在一示例中,建立一个轻量级的事务链路日志处理系统。无论是否是确定顺序的事务链路,都预先对分支事务进行顺序编号。将分支事务Local Tx1、Local Tx2、Local Tx3、Local Tx4和Local Tx5分别编号为1、2、3、4和5。每个分支事务执行完成,发送顺序标记日志,顺序标记日志记录了分支事务的事务标识。触发回滚时,由事务链路日志处理系统根据已经保存的事务链路日志,判断分支事务的实际执行顺序是否与预先确定的执行顺序相匹配,即判断是否是确定顺序的事务链路。若当前全局事务的分支事务的日志编号是连续的,则为确定顺序的事务链路;否则不是。例如,如果Local Tx1、Local Tx2、Local Tx3、Local Tx4、Local Tx5执行之后,获取到顺序标记日志确定的执行循序是1、2、3、5、4,则不是确定顺序的事务链路。如果获取到顺序标记日志确定的执行循序是1、2、3、4、5,则是确定顺序的事务链路。如果是确定顺序的事务链路,则按照实际执行顺序倒序进行回滚,否者,启动流程引擎进行事务回滚。

[0053] 与本申请实施例提供的方法的应用场景以及方法相对应地,本申请实施例还提供一种分布式事务回滚装置。如图4所示为本申请一实施例的分布式事务回滚装置的结构框图,该装置包括:

[0054] 获取模块401,用于在监听到全局事务的分支事务的回滚事务事件时,获取所述全局事务的多个已执行的分支事务的实际执行顺序信息。

[0055] 比对模块402,用于若多个已执行的分支事务的实际执行顺序信息与预先确定的全局事务的多个分支事务的执行顺序信息匹配,则将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比对;事务标识是根据多个已执行的分支事务的实际执行顺序确定的。

[0056] 回滚模块403,用于根据比对结果,在多个已执行的分支事务中依次回滚到最先执行的分支事务。

[0057] 本申请提供了一种分布式事务回滚装置,在监听到全局事务的分支事务的回滚事务事件时,获取全局事务的多个已执行的分支事务的实际执行顺序信息;若多个已执行的分支事务的实际执行顺序信息与预先确定的全局事务的多个分支事务的执行顺序信息匹配,则将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比对;事务标识是根据多个已执行的分支事务的实际执行顺序确定的;根据比对结果,在多个已执行的分支事务中依次回滚到最先执行的分支事务。本实施例中,对于按照确定的执行顺序执行的分支事务,即该分支事务的实际执行顺序与预先确定的执行顺序相匹配,则通过将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比对,实现已执行的分支事务的回滚,采用基于回滚事务事件驱动的方式,不需要引入流程引擎,可以降低事务回滚的实现成本。

[0058] 在一种实现方式中,事务标识为事务顺序编号,比对模块402在将回滚事务事件中的事务标识与多个已执行的分支事务的事务标识进行比对时,用于:确定回滚事务事件中的事务顺序编号与多个已执行的分支事务的事务顺序编号之间的差值。

[0059] 在一种实现方式中,回滚模块403,用于:若回滚事务事件中的事务顺序编号与已执行的分支事务的事务顺序编号之间的差值为预设值,则将已执行的分支事务确定为目标分支事务;从多个目标分支事务依次回滚到多个已执行的分支事务中最先执行的分支事务。

[0060] 在一种实现方式中,回滚模块403在从多个目标分支事务依次回滚到多个已执行的分支事务中最先执行的分支事务时,用于:回滚到目标分支事务,生成目标分支事务的回滚事务事件;根据当前的目标分支事务的回滚事务事件,回滚到下一目标分支事务;从多个目标分支事务,依次回滚到多个已执行的分支事务中最先执行的分支事务。

[0061] 在一种实现方式中,回滚模块403在根据当前的目标分支事务的回滚事务事件,回滚到下一目标分支事务时,用于:

[0062] 根据当前的目标分支事务的回滚事务事件中的事务顺序编号,确定下一目标分支事务;

[0063] 根据当前的目标分支事务的回滚事务事件中的上下文信息,回滚到下一目标分支事务。

[0064] 在一种实现方式中,回滚模块403在从多个目标分支事务,依次回滚到多个已执行的分支事务中最先执行的分支事务时,还用于:

[0065] 从当前的目标分支事务依次回滚到多个已执行的分支事务中最先执行的分支事务,并生成最先执行的分支事务的回滚事务事件。

[0066] 在一种实现方式中,获取模块401,用于:

[0067] 获取全局事务的多个已执行的分支事务分别对应的顺序标记日志;分支事务对应的顺序标记日志为分支事务执行完成时生成的;

[0068] 根据多个顺序标记日志,确定多个已执行的分支事务的实际执行顺序信息。

[0069] 在一种实现方式中,装置还用于:

[0070] 若多个已执行的分支事务的实际执行顺序信息与预先确定的全局事务的多个分支事务的执行顺序信息不匹配,则启动流程引擎,利用流程引擎驱动全局事务的分支事务进行事务回滚。

[0071] 在一种实现方式中,分支事务包括微服务。

[0072] 本申请实施例各装置中的各模块的功能可以参见上述方法中的对应描述,并具备相应的有益效果,在此不再赘述。

[0073] 图5为用来实现本申请实施例的电子设备的框图。如图5所示,该电子设备包括:存储器510和处理器520,存储器510内存储有可在处理器520上运行的计算机程序。处理器520执行该计算机程序时实现上述实施例中的方法。存储器510和处理器520的数量可以为一个或多个。

[0074] 该电子设备还包括:

[0075] 通信接口530,用于与外界设备进行通信,进行数据交互传输。

[0076] 如果存储器510、处理器520和通信接口530独立实现,则存储器510、处理器520和通信接口530可以通过总线相互连接并完成相互间的通信。该总线可以是工业标准体系结构(Industry Standard Architecture,ISA)总线、外部设备互连(Peripheral Component Interconnect,PCI)总线或扩展工业标准体系结构(Extended Industry Standard Architecture,EISA)总线等。该总线可以分为地址总线、数据总线、控制总线等。为便于表示,图5中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。

[0077] 可选的,在具体实现上,如果存储器510、处理器520及通信接口530集成在一块芯片上,则存储器510、处理器520及通信接口530可以通过内部接口完成相互间的通信。

[0078] 本申请实施例提供了一种计算机可读存储介质,其存储有计算机程序,该程序被处理器执行时实现本申请实施例中提供的方法。

[0079] 本申请实施例还提供了一种芯片,该芯片包括处理器,用于从存储器中调用并运行存储器中存储的指令,使得安装有芯片的通信设备执行本申请实施例提供的方法。

[0080] 本申请实施例还提供了一种芯片,包括:输入接口、输出接口、处理器和存储器,输入接口、输出接口、处理器以及存储器之间通过内部连接通路相连,处理器用于执行存储器中的代码,当代码被执行时,处理器用于执行申请实施例提供的方法。

[0081] 应理解的是,上述处理器可以是中央处理器(Central Processing Unit,CPU),还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现场可编程门阵列(Field Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者是任何常规的处理器等。值得说明的是,处理器可以是支持进阶精简指令集机器(Advanced RISC Machines,ARM)架构的处理器。

[0082] 进一步地,可选的,上述存储器可以包括只读存储器和随机访问存储器。该存储器可以是易失性存储器或非易失性存储器,或可包括易失性和非易失性存储器两者。其中,非易失性存储器可以包括只读存储器(Read-Only Memory,ROM)、可编程只读存储器(Programmable ROM,PROM)、可擦除可编程只读存储器(Erasable PROM,EPROM)、电可擦除可编程只读存储器(Electrically EPROM,EEPROM)或闪存。易失性存储器可以包括随机访问存储器(Random Access Memory,RAM),其用作外部高速缓存。通过示例性但不是限制性说明,许多形式的RAM均可用。例如,静态随机访问存储器(Static RAM,SRAM)、动态随机访问存储器(Dynamic Random Access Memory,DRAM)、同步动态随机访问存储器(Synchronous DRAM,SDRAM)、双倍数据速率同步动态随机访问存储器(Double Data Rate SDRAM,DDR SDRAM)、增强型同步动态随机访问存储器(Enhanced SDRAM,ESDRAM)、同步链接动态随机访问存储器(Sync link DRAM,SLDRAM)和直接内存总线随机访问存储器(Direct Rambus RAM,DR RAM)。

[0083] 在上述实施例中,可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时,可以全部或部分地以计算机程序产品的形式实现。计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行计算机程序指令时,全部或部分地产生依照本申请的流程或功能。计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。计算机指令可以存储在计算机可读存储介质中,或者从一个计算机可读存储介质向另一个计算机可读存储介质传输。

[0084] 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、材料或者特点包括于本申请的至少一个实施例或示例中。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。

[0085] 此外,术语“第一”、“第二”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”的特征可以明示或隐

含地包括至少一个该特征。在本申请的描述中，“多个”的含义是两个或两个以上，除非另有明确具体的限定。

[0086] 流程图中描述的或在此以其他方式描述的任何过程或方法可以被理解为，表示包括一个或多个用于实现特定逻辑功能或过程的步骤的可执行指令的代码的模块、片段或部分。并且本申请的优选实施方式的范围包括另外的实现，其中可以不按所示出或讨论的顺序，包括根据所涉及的功能按基本同时的方式或按相反的顺序，来执行功能。

[0087] 在流程图中描述的或在此以其他方式描述的逻辑和/或步骤，例如，可以被认为是用于实现逻辑功能的可执行指令的定序列表，可以具体实现在任何计算机可读介质中，以供指令执行系统、装置或设备（如基于计算机的系统、包括处理器的系统或其他可以从指令执行系统、装置或设备取指令并执行指令的系统）使用，或结合这些指令执行系统、装置或设备而使用。

[0088] 应理解的是，本申请的各部分可以用硬件、软件、固件或它们的组合来实现。在上述实施方式中，多个步骤或方法可以用存储在存储器中且由合适的指令执行系统执行的软件或固件来实现。上述实施例方法的全部或部分步骤是可以通程序来指令相关的硬件完成，该程序可以存储于一种计算机可读存储介质中，该程序在执行时，包括方法实施例的步骤之一或其组合。

[0089] 此外，在本申请各个实施例中的各功能单元可以集成在一个处理模块中，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模块中。上述集成的模块既可以采用硬件的形式实现，也可以采用软件功能模块的形式实现。上述集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用，也可以存储在一个计算机可读存储介质中。该存储介质可以是只读存储器，磁盘或光盘等。

[0090] 以上所述，仅为本申请的示例性实施方式，但本申请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申请记载的技术范围内，可轻易想到其各种变化或替换，这些都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应以权利要求的保护范围为准。

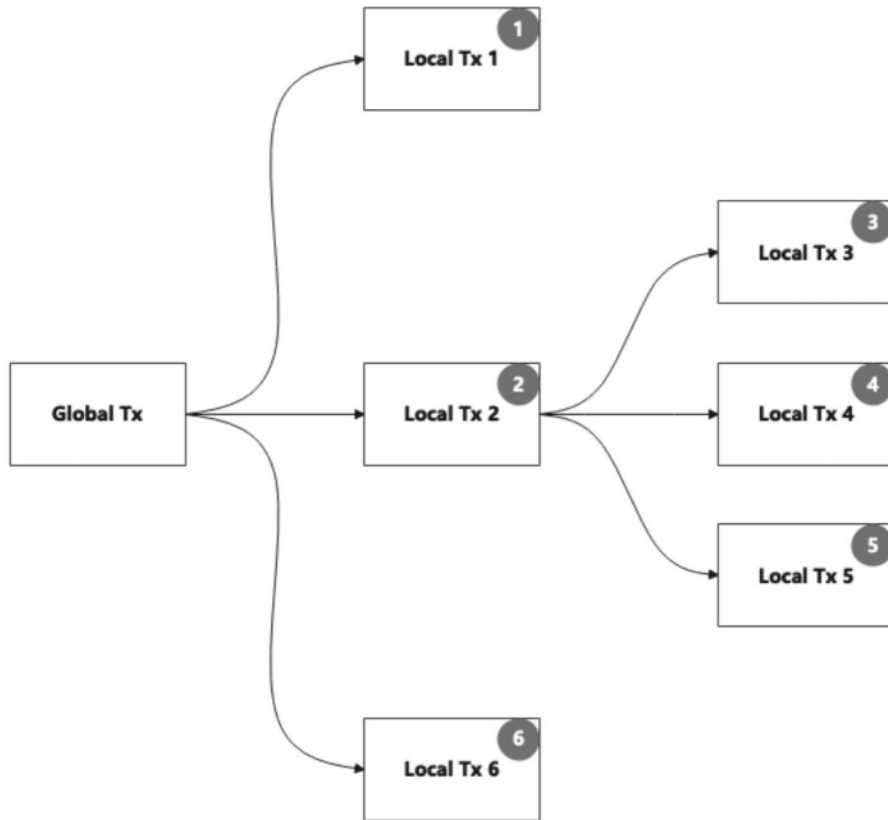


图1

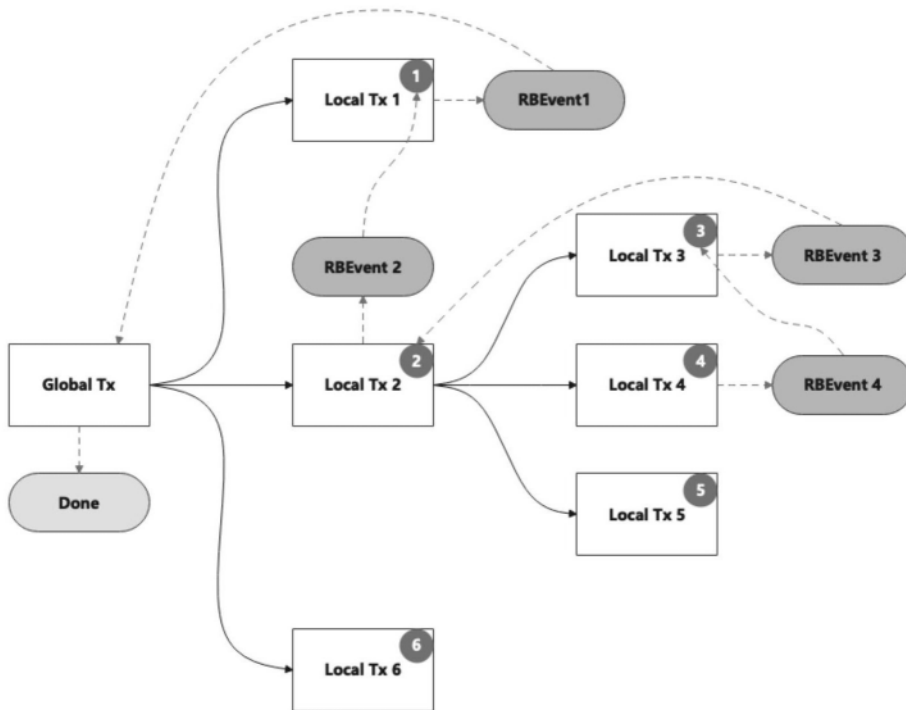


图2

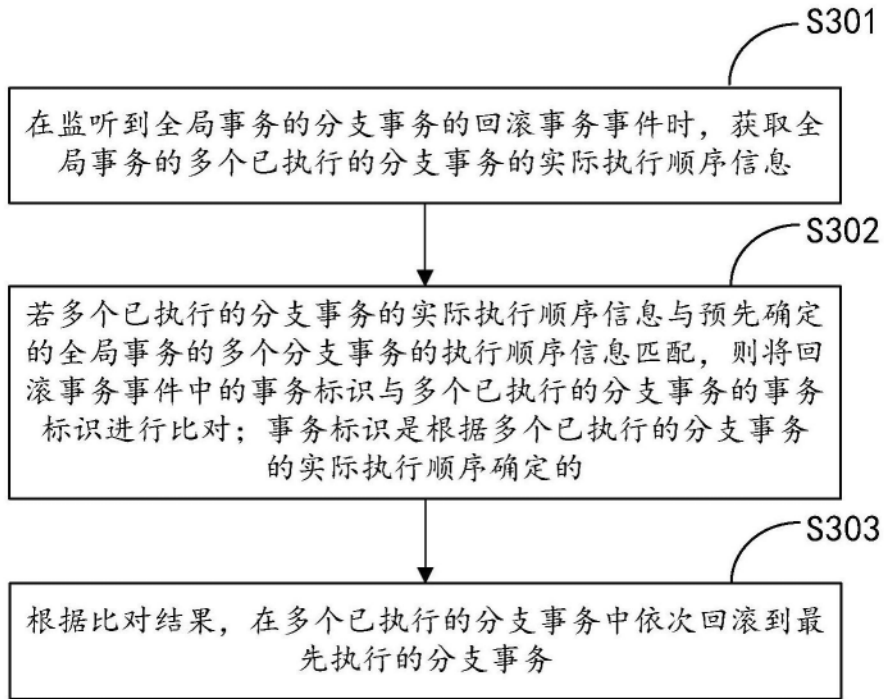


图3

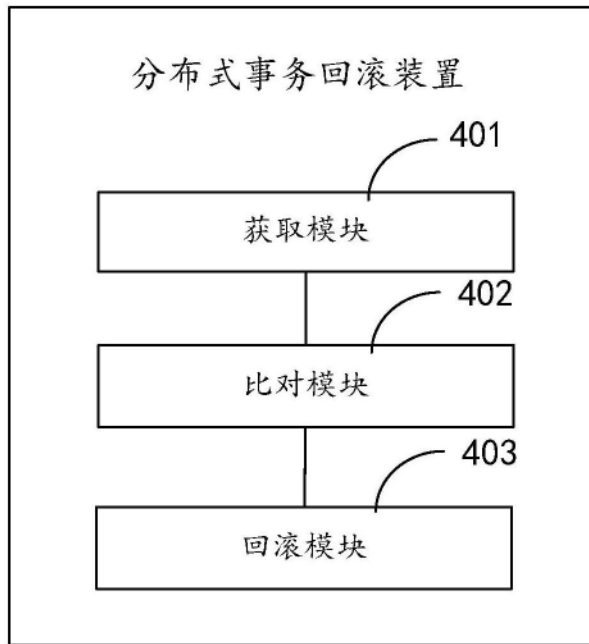


图4

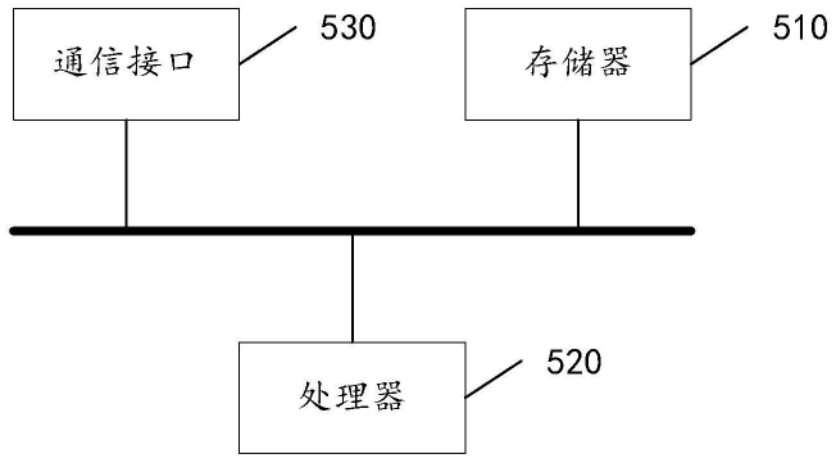


图5