(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 117874145 B
(45)授权公告日 2024.05.28

(21)申请号 202410285023.5
(22)申请日 2024.03.13
(65)同一申请的已公布的文献号
申请公布号 CN 117874145 A
(43)申请公布日 2024.04.12
(73)专利权人 连连(杭州)信息技术有限公司
地址 310000 浙江省杭州市滨江区越达巷79号1号楼12楼A-1
(72)发明人 王愚 郭进 初永光 陈俊维
(74)专利代理机构 广州三环专利商标代理有限公司 44202
专利代理师 黄盼
(51)Int.Cl.
G06F 16/27(2019.01)
G06F 16/23(2019.01)
(56)对比文件
US 2015347547 A1,2015.12.03
US 2011099420 A1,2011.04.28
CN 112486718 A,2021.03.12
CN 108319617 A,2018.07.24
US 2024004902 A1,2024.01.04
CN 104750755 A,2015.07.01
US 2007233699 A1,2007.10.04
CN 114691771 A,2022.07.01
CA 2545532 A1,2006.11.04
CN 104618127 A,2015.05.13
CN 107015885 A,2017.08.04
CN 107357800 A,2017.11.17
CN 111930465 A,2020.11.13
CN 111966699 A,2020.11.20
CN 112685234 A,2021.04.20
CN 113141370 A,2021.07.20
CN 114493097 A,2022.05.13
CN 115421976 A,2022.12.02
CN 117555966 A,2024.02.13
JP 2006311064 A,2006.11.09 (续)
审查员 吴海旋

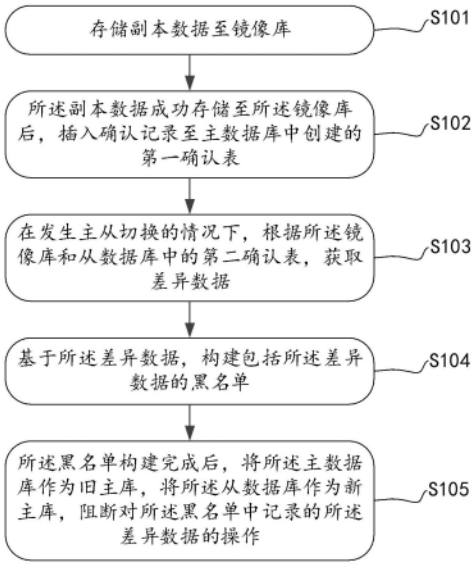
权利要求书2页 说明书12页 附图4页

(54)发明名称

一种主从数据库的强一致方法、装置、设备及存储介质

(57)摘要

本公开涉及一种主从数据库的强一致方法、装置、设备及存储介质，上述方法包括存储副本数据至镜像库；上述副本数据成功存储至上述镜像库后，插入确认记录至主数据库中创建的第一确认表；在发生主从切换的情况下，根据上述镜像库和从数据库中的第二确认表，获取差异数据；基于上述差异数据，构建包括上述差异数据的黑名单；上述黑名单构建完成后，将上述从数据库作为新主库，阻断对上述黑名单中记录的上述差异数据的操作。本公开可以确保数据在主从数据库间的一致性，减少业务中断时间，提升业务连续性；阻断对黑名单中数据的操作，有效避免对不一致数据的访问和修改，降低因数据不一致带来的业务风险，提高数据操作的安全性。



CN 117874145 B

[转续页]

[接上页]

(56)对比文件

US 11537314 B1,2022.12.27

US 2006026452 A1,2006.02.02

US 2022019575 A1,2022.01.20

王锦;梁正和;王法强.表广播机制在MyCat中的实现.计算机技术与发展.2017,(03),全文.

李旭风.打造与“两地三中心”相适应的应用架构体系.中国金融电脑.2015,(第09期),全文.

于翔;叶德建.一种基于云的应用层容错机制设计与实现.微型电脑应用.2016,(第02期),全文.

宁泰安;刘金刚.基于SQL-92和JDBC的SpaceOS数据库数据迁移的实现.计算机应用与软件.2014,(12),全文.

1.一种主从数据库的强一致方法,其特征在于,包括:

存储副本数据至镜像库;所述副本数据的获取过程包括:拦截事务执行过程中的数据写操作语句;基于预设规则,提取所述数据写操作语句中的核心数据,所述核心数据包括所述数据写操作语句中的表名、关键字段名和所述关键字段名对应的值,所述表名和所述关键字段名均属于所述预设规则中定义的数据;存储特定格式的所述核心数据至事务上下文;生成事务唯一标识符,所述事务唯一标识符用于标识和追踪对应的事务;结合所述事务唯一标识符和所述事务上下文中存储的所述核心数据,得到所述副本数据;

所述副本数据成功存储至所述镜像库后,插入确认记录至主数据库中创建的第一确认表;所述第一确认表包括所述事务唯一标识符字段和时间戳字段,所述时间戳字段用于记录插入所述确认记录的时间;

在发生主从切换的情况下,根据所述镜像库和从数据库中的第二确认表,获取差异数据,所述第二确认表是基于主从复制机制创建在所述从数据库中的表;

基于所述差异数据,构建包括所述差异数据的黑名单;

所述黑名单构建完成后,将所述主数据库作为旧主库,将所述从数据库作为新主库,阻断对所述黑名单中记录的所述差异数据的操作。

2.根据权利要求1所述的一种主从数据库的强一致方法,其特征在于,所述在发生主从切换的情况下,根据所述镜像库和从数据库中的第二确认表,获取差异数据,包括:

所述在发生主从切换的情况下,对比所述镜像库和所述第二确认表,确定所述第二确认表中未记录的事务唯一标识符;

基于所述未记录的事务唯一标识符,获取所述差异数据,所述差异数据为所述镜像库中,所述未记录的事务唯一标识符对应的所述核心数据。

3.根据权利要求1所述的一种主从数据库的强一致方法,其特征在于,所述方法还包括:

所述旧主库恢复后,补齐所述旧主库和所述新主库缺失的数据;

释放所述黑名单中的数据。

4.根据权利要求1所述的一种主从数据库的强一致方法,其特征在于,所述在发生主从切换的情况之前,所述方法还包括:

在所述确认记录成功插入所述主数据库中创建的第一确认表的情况下,提交所述事务;

在所述确认记录未成功插入所述主数据库中创建的第一确认表的情况下,所述事务回滚至初始状态。

5.一种主从数据库的强一致装置,其特征在于,包括:

存储模块,用于存储副本数据至镜像库;所述副本数据的获取过程包括:拦截事务执行过程中的数据写操作语句;基于预设规则,提取所述数据写操作语句中的核心数据,所述核心数据包括所述数据写操作语句中的表名、关键字段名和所述关键字段名对应的值,所述表名和所述关键字段名均属于所述预设规则中定义的数据;存储特定格式的所述核心数据至事务上下文;生成事务唯一标识符,所述事务唯一标识符用于标识和追踪对应的事务;结合所述事务唯一标识符和所述事务上下文中存储的所述核心数据,得到所述副本数据;

插入模块,用于所述副本数据成功存储至所述镜像库后,插入确认记录至主数据库中

创建的第一确认表;所述第一确认表包括所述事务唯一标识符字段和时间戳字段,所述时间戳字段用于记录插入所述确认记录的时间;

获取模块,用于在发生主从切换的情况下,根据所述镜像库和从数据库中的第二确认表,获取差异数据,所述第二确认表是基于主从复制机制创建在所述从数据库中的表;

构建模块,用于基于所述差异数据,构建包括所述差异数据的黑名单;

阻断模块,用于所述黑名单构建完成后,将所述主数据库作为旧主库,将所述从数据库作为新主库,阻断对所述黑名单中记录的所述差异数据的操作。

6.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质中存储有至少一条指令或至少一段程序,所述至少一条指令或至少一段程序由处理器加载并执行以实现如权利要求1-4中任意一项所述的一种主从数据库的强一致方法。

7.一种电子设备,其特征在于,包括至少一个处理器,以及与所述至少一个处理器通信连接的存储器;其中,所述存储器存储有可被所述至少一个处理器执行的指令,所述至少一个处理器通过执行所述存储器存储的指令实现如权利要求1-4中任意一项所述的一种主从数据库的强一致方法。

一种主从数据库的强一致方法、装置、设备及存储介质

技术领域

[0001] 本公开涉及互联网技术领域,尤其涉及一种主从数据库的强一致方法、装置、设备及存储介质。

背景技术

[0002] 分布式存储系统由多个存储节点构成,每个节点负责存储一部分数据,节点通过网络相互连接并协同工作,提供高效率和高可靠性的数据服务。在这种系统中,数据可以分散存储于不同的物理位置,而逻辑上仍然作为一个整体被管理和访问。

[0003] 在分布式存储系统中,主从数据库架构是实现数据管理和一致性的一种方式。该架构通常包括一个主数据库和至少一个从数据库,主数据库负责处理所有写入操作以及维护数据的最新状态和完整性,而从数据库负责同步主数据库的数据变更并处理读请求。主从数据库架构不仅允许在多个节点间分散数据存储和处理负载,还通过复制和同步机制确保了数据在整个系统中的一致性。

[0004] 主从复制机制是实现主从数据库架构的关键,通过同步数据变更来维护数据一致性。主数据库和从数据库能够有效地协同工作,保持数据的一致性和最新状态。在主从复制机制中,主数据库在处理任何写入操作时,会将变更记录到一个特定的日志文件中,通常是二进制日志,从数据库通过定期拉取并应用这些日志记录来同步数据,确保自身的数据副本与主数据库保持一致。然而,主从复制机制并不是强一致性协议,在网络延迟或故障时,从数据库中的数据可能会滞后于主数据库,导致数据不一致的问题。尽管主从数据库架构为了增强数据的一致性引入了半同步和全同步复制机制,但是由于网络问题,半同步复制机制并不能完全消除数据不一致的风险,可能导致在主从切换时出现数据不一致,从而引发生产问题,而全同步复制机制可能会由于网络延迟问题使主数据库陷入阻塞状态,影响整个系统的可用性。

发明内容

[0005] 为了解决上述提出的至少一个技术问题,本公开提出了一种主从数据库的强一致方法、装置、设备及存储介质。

[0006] 根据本公开实施例的第一方面,提供一种主从数据库的强一致方法,包括:

[0007] 存储副本数据至镜像库;

[0008] 所述副本数据成功存储至所述镜像库后,插入确认记录至主数据库中创建的第一确认表;

[0009] 在发生主从切换的情况下,根据所述镜像库和从数据库中的第二确认表,获取差异数据,所述第二确认表是基于主从复制机制创建在所述从数据库中的表;

[0010] 基于所述差异数据,构建包括所述差异数据的黑名单;

[0011] 所述黑名单构建完成后,将所述主数据库作为旧主库,将所述从数据库作为新主库,阻断对所述黑名单中记录的所述差异数据的操作。

[0012] 在一个实施例中,所述存储副本数据至镜像库之前,所述方法还包括:

[0013] 拦截事务执行过程中的数据写操作语句;

[0014] 基于预设规则,提取所述数据写操作语句中的核心数据,所述核心数据包括所述数据写操作语句中的表名、关键字段名和所述关键字段名对应的值;所述表名和所述关键字段名均属于所述预设规则中定义的数据;

[0015] 存储特定格式的所述核心数据至事务上下文。

[0016] 在一个实施例中,所述存储特定格式的所述核心数据至事务上下文之后,存储副本数据至镜像库之前,所述方法还包括:

[0017] 生成事务唯一标识符,所述事务唯一标识符用于标识和追踪对应的事务;

[0018] 结合所述事务唯一标识符和所述事务上下文中存储的所述核心数据,得到所述副本数据。

[0019] 在一个实施例中,所述副本数据成功存储至所述镜像库后,插入确认记录至主数据库中创建的第一确认表,包括:

[0020] 在所述主数据库中创建所述第一确认表,所述第一确认表包括所述事务唯一标识符字段和时间戳字段,所述时间戳字段用于记录插入所述确认记录的时间;

[0021] 所述副本数据成功存储至所述镜像库后,插入所述事务唯一标识符和所述时间戳至所述第一确认表。

[0022] 在一个实施例中,所述在发生主从切换的情况下,根据所述镜像库和从数据库中的第二确认表,获取差异数据,包括:

[0023] 所述在发生主从切换的情况下,对比所述镜像库和所述第二确认表,确定所述第二确认表中未记录的事务唯一标识符;

[0024] 基于所述未记录的事务唯一标识符,获取所述差异数据,所述差异数据为所述镜像库中,所述未记录的事务唯一标识符对应的所述核心数据。

[0025] 在一个实施例中,所述方法还包括:

[0026] 所述旧主库恢复后,补齐所述旧主库和所述新主库缺失的数据;

[0027] 释放所述黑名单中的数据。

[0028] 在一个实施例中,所述在发生主从切换的情况之前,所述方法还包括:

[0029] 在所述确认记录成功插入所述主数据库中创建的第一确认表的情况下,提交所述事务;

[0030] 在所述确认记录未成功插入所述主数据库中创建的第一确认表的情况下,所述事务回滚至初始状态。

[0031] 根据本公开实施例的第二方面,提供一种主从数据库的强一致装置,包括:

[0032] 存储模块,用于存储副本数据至镜像库;

[0033] 插入模块,用于所述副本数据成功存储至所述镜像库后,插入确认记录至主数据库中创建的第一确认表;

[0034] 获取模块,用于在发生主从切换的情况下,根据所述镜像库和从数据库中的第二确认表,获取差异数据,所述第二确认表是基于主从复制机制创建在所述从数据库中的表;

[0035] 构建模块,用于基于所述差异数据,构建包括所述差异数据的黑名单;

[0036] 阻断模块,用于所述黑名单构建完成后,将所述主数据库作为旧主库,将所述从数

据库作为新主库,阻断对所述黑名单中记录的所述差异数据的操作。

[0037] 根据本公开实施例的第三方面,提供一种计算机可读存储介质,所述计算机可读存储介质中存储有至少一条指令或至少一段程序,所述至少一条指令或至少一段程序由处理器加载并执行以实现如上述第一方面所述的一种主从数据库的强一致方法。

[0038] 根据本公开实施例的第四方面,提供一种电子设备,包括至少一个处理器,以及与所述至少一个处理器通信连接的存储器;其中,所述存储器存储有可被所述至少一个处理器执行的指令,所述至少一个处理器通过执行所述存储器存储的指令实现如上述第一方面所述的一种主从数据库的强一致方法。

[0039] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,而非限制本公开。

[0040] 实施本公开,具有以下有益效果:

[0041] 存储副本数据至镜像库中,确保数据在主从数据库间的完整性;使用确认表记录成功复制到镜像库的事务,并在主从切换时比对镜像库与从数据库中的确认表获取差异数据,能够确保数据在主从数据库间的一致性;在主从切换后将从数据库提升为新主库,减少业务中断时间,提升业务的连续性和用户的体验感;构建黑名单并在新主库生效后阻断对黑名单中数据的操作,能够有效避免对不一致数据的访问和修改,降低因数据不一致带来的业务风险,提高数据操作的安全性;与半同步和全同步复制机制相比,在部署和维护上更加简单和轻量。

[0042] 根据下面参考附图对示例性实施例的详细说明,本公开的其它特征及方面将变得清楚。

附图说明

[0043] 为了更清楚地说明本说明书实施例或现有技术中的技术方案和优点,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的介绍,显而易见地,下面描述中的附图仅仅是本说明书的一些实施例,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它附图。

[0044] 图1示出根据本公开实施例的一种主从数据库的强一致方法的流程示意图。

[0045] 图2示出根据本公开实施例的获取副本数据的流程示意图。

[0046] 图3示出根据本公开实施例的事务提交的流程示意图。

[0047] 图4示出根据本公开实施例的获取差异数据的流程示意图。

[0048] 图5示出根据本公开实施例的一种主从数据库的强一致装置的模块示意图。

[0049] 图6示出根据本公开实施例的一种电子设备的框图。

具体实施方式

[0050] 下面将结合本说明书实施例中的附图,对本说明书实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本说明书一部分实施例,而不是全部的实施例。基于本说明书中的实施例,本领域普通技术人员在没有做出创造性劳动的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。

[0051] 需要说明的是,本发明的说明书和权利要求书及上述附图中的术语“第一”、“第

二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的本发明的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,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、产品或服务器不必限于清楚地列出的那些步骤或单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0052] 以下将参考附图详细说明本公开的各种示例性实施例、特征和方面。附图中相同的附图标记表示功能相同或相似的元件。尽管在附图中示出了实施例的各种方面,但是除非特别指出,不必按比例绘制附图。

[0053] 在这里专用的词“示例性”意为“用作例子、实施例或说明性”。这里作为“示例性”所说明的任何实施例不必解释为优于或好于其它实施例。

[0054] 本文中术语“和/或”,仅仅是一种描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。另外,本文中术语“至少一种”表示多种中的任意一种或多种中的至少两种的任意组合,例如,包括A、B、C中的至少一种,可以表示包括从A、B和C构成的集合中选择的任意一个或多个元素。

[0055] 另外,为了更好地说明本公开,在下文的具体实施方式中给出了众多的具体细节。本领域技术人员应当理解,没有某些具体细节,本公开同样可以实施。在一些实例中,对于本领域技术人员熟知的方法、手段、元件和电路未作详细描述,以便于凸显本公开的主旨。

[0056] 在金融领域,数据的一致性、完整性和系统的可靠性至关重要。半同步复制通过确保每个事务至少被复制到一个从数据库中,一定程度上加强了数据的一致性。然而,网络的不稳定性或从数据库的故障可能在主从切换过程中导致数据不一致,在金融交易中可能引起严重后果,如重复交易、交易遗漏或错误的交易金额。全同步复制虽然提供了数据一致性保障,然而在网络延迟较高或从数据库性能不佳的情况下,可能降低系统的整体可用性。由于主数据库需要等待所有从数据库的确认,可能导致主数据库阻塞,使整个系统无法响应用户请求,进而影响交易的执行速度和效率。

[0057] 图1示出根据本公开实施例的一种主从数据库的强一致方法的流程示意图,如图1,上述方法包括:

[0058] S101、存储副本数据至镜像库。

[0059] 图2示出根据本公开实施例的获取副本数据的流程示意图,如图2,获取副本数据包括:

[0060] S201、拦截事务执行过程中的数据写操作语句。

[0061] 选择合适的拦截技术,在事务开始后,拦截所有发往数据库的数据写操作语句。本公开并不对拦截技术进行限定,在一个实施例中,采用Java数据库连接(Java Database Connectivity,JDBC)驱动层拦截SQL写语句。JDBC驱动层是Java应用程序与数据库之间交互的中介,负责传递Java应用程序中的SQL命令至数据库,并将数据库的执行结果回传至Java应用程序。通过在JDBC驱动层实施拦截,能够在Java应用程序将SQL命令提交至数据库执行之前介入,实现对所有经由JDBC层的SQL写语句的截获。

[0062] 当在JDBC驱动层实施拦截时,可以采用Java动态代理机制或自定义包装器两种主要技术。

[0063] Java动态代理机制允许在应用程序运行时为特定接口动态生成实现类的代理对象,适用于如Connection、Statement等JDBC核心接口。代理对象的创建是通过实现InvocationHandler接口完成的,当代理对象的任何方法被调用时,调用将被转发到关联的InvocationHandler的invoke方法。在JDBC的应用场景中,例如,当应用程序通过代理的Statement对象执行executeUpdate()方法进行SQL写操作时,将触发代理对象关联的InvocationHandler的invoke方法。在invoke方法内部,拦截逻辑得以执行。invoke方法中可以嵌入SQL解析工具,如JSQLParser或ANTLR,以解析传递给executeUpdate()方法的SQL语句,根据预设的逻辑进行相应的处理。

[0064] 自定义包装器可以创建自定义的Connection、Statement包装类,重写执行SQL语句的方法,如executeUpdate()方法等,在上述方法中加入拦截逻辑,根据预设的逻辑进行相应的处理。

[0065] S202、基于预设规则,提取所述数据写操作语句中的核心数据,所述核心数据包括所述数据写操作语句中的表名、关键字段名和所述关键字段名对应的值;所述表名和所述关键字段名均属于所述预设规则中定义的数据。

[0066] 在JDBC驱动层拦截到数据写操作语句后,对其进行解析,提取出表名、字段名及其相应值等关键组成部分,为后续的数据处理与决策提供了基础信息。对数据写操作语句进行解析之后,基于预设规则,提取数据写操作语句中的核心数据。本公开并不对配置预设规则的方法进行限定,在一个实施例中,利用Apollo配置预设规则。Apollo是一个开源的分布式配置中心,主要用于集中管理应用的配置信息。Apollo提供了配置的实时更新、版本变更历史、灰度发布等功能,并能够适用于微服务架构。通过在Apollo中配置拦截规则,例如指定特定的表和字段进行拦截,能够实现无需更改代码即可更新规则。当业务需求变更,例如需要监控新的表和字段,或者停止拦截某些表和字段时,只需在Apollo中更新配置,无需修改任何代码即可实现规则的动态调整。本公开并不对预设规则进行限定,规则根据实际需求设计,在一个实施例中,例如账户系统,该系统涉及账务表和流水表,账务表和流水表都有用于标识账户的账户号字段,账务系统需要以"账户号"作为关键维度进行更新操作,因此可将账务表表名、流水表表名、账户号字段名配置在Apollo的规则中。根据Apollo中配置的规则,从拦截的数据写操作语句中提取预设规则所指定的表名和字段名,以及与字段名对应的值。通过将Apollo与JDBC驱动层拦截结合使用,可以构建一个既灵活又高效的数据拦截与处理机制,显著提升了数据监控和管理的能力。

[0067] S203、存储特定格式的所述核心数据至事务上下文。

[0068] 在应用程序中,定义事务上下文作为一个数据结构,可以是一个类或结构体,用于在整个事务执行过程中持有和传递关键信息。事务上下文可以包含多个与该事务相关的信息,如事务ID、执行时间、涉及的表和字段信息等。上述根据预设规则提取的核心数据,需要被格式化为事务上下文可以接受的形式。本公开并不对核心数据在事务上下文中的具体格式进行限定,在一个实施例中,核心数据可以被格式化为[{表名:{字段名:[字段值...]}},...]的结构。在Java中,可以通过使用List、Map等集合类型实现上述结构。

[0069] 通过拦截事务执行过程中的数据写操作语句并基于预设规则提取核心数据,能够精确地识别和记录被修改的数据,包括相关表名、关键字段名及其对应的值等信息,确保了对数据变更的详细追踪;存储特定格式的核心数据至事务上下文中,为后续的一致性检查

提供了关键信息;增强了在主从切换时的数据处理准确性和效率,从而保障了整个系统的数据一致性和可靠性。

[0070] S204、生成事务唯一标识符,所述事务唯一标识符用于标识和追踪对应的事务。

[0071] 本公开并不对事务唯一标识符的生成方式进行限定,在一个实施例中,事务唯一标识符为用于计算机系统中以确保全局唯一性的标识符(Universally Unique Identifier,UUID),事务唯一标识符由应用层生成,通常采用UUID算法,在Java中可以直接使用java.util.UUID类生成一个UUID。事务唯一标识符为每个事务提供了不可重复的识别码,确保每个事务的可追溯性和可管理性。

[0072] S205、结合所述事务唯一标识符和所述事务上下文中存储的所述核心数据,得到副本数据。通过将事务唯一标识符与核心数据结合,确保每次数据变更都被准确记录和追踪,从而有效地维护数据的完整性和一致性。

[0073] 镜像库是分布式存储系统的一个组成部分,用于存储数据的副本。将副本数据存储至镜像库,通过记录每次事务的详细变更,可以确保数据在主从数据库间的完整性和一致性。

[0074] 将上述副本数据存储至镜像库中,首先,需要选择一个数据库系统作为镜像库,本公开并不对镜像库进行限定,镜像库的选择取决于多种因素,包括现有的技术栈、性能需求、可靠性、可维护性等,在一个实施例中,选择MySQL数据库作为镜像库;确定使用的数据库后,需要涉及存储副本数据的表,本公开并不对用于存储副本数据的表的具体结构进行限定,在一个实施例中,表不仅用于存储副本数据,还包含其他额外的字段以满足广泛的业务需求。表具体设计如下:

[0075] CREATE TABLE `example_prepare` (

[0076] `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT,

[0077] `uid` bigint(20) NOT NULL COMMENT,

[0078] `source_key` varchar(100) NOT NULL COMMENT,

[0079] `data` text NOT NULL COMMENT,

[0080] `status` tinyint(1) DEFAULT NULL COMMENT,

[0081] `create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT,

[0082] `update_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT,

[0083] PRIMARY KEY (`id`),

[0084] UNIQUE key `uidx_uid` (`uid`),

[0085] KEY `idx_create_time_source_key` (`create_time`,`source_key`)

[0086]) ENGINE=InnoDBDEFAULT CHARSET=utf8mb4 ROW_FORMAT=DYNAMIC;

[0087] 上述存储副本数据的表的字段包括:

[0088] 主键(id):唯一标识表中的每一行记录;

[0089] 事务唯一标识符字段(uid):存储每个事务对应的事务唯一标识符;

[0090] 来源标识字段(source_key):存储来源标识,例如可以是触发事务的主数据库的IP地址;

[0091] 数据字段(data):存储特定格式的核心数据,例如序列化的JSON格式的数据{"

test":{"account_no":["123456"]}};

[0092] 状态字段(status):记录该行记录的状态,例如0表示准备状态,2表示回滚状态等;

[0093] 创建时间字段(create_time):时间戳类型,记录数据创建的时间;

[0094] 更新时间字段(update_time):时间戳类型,记录数据最后一次更新的时间。

[0095] S102、所述副本数据成功存储至所述镜像库后,插入确认记录至主数据库中创建的第一确认表。

[0096] 在主数据库中创建第一确认表,第一确认表的创建可以由数据库管理员手动执行,或者设计为在应用程序启动时自动检测并创建。本公开并不对第一确认表进行限定,在一个实施例中,第一确认表包括事务唯一标识符字段和时间戳字段,时间戳字段用于记录插入确认记录的时间。在一个实施例中,时间戳字段包括创建时间字段(create_time)和更新时间字段(update_time),创建时间字段用于记录确认记录的创建时间,更新时间字段用于记录确认记录最后更新时间,第一确认表具体设计如下:

[0097] CREATE TABLE `dama_ack` (

[0098] `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT,

[0099] `uid` bigint(20) NOT NULL COMMENT,

[0100] `create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT,

[0101] `update_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT,

[0102] PRIMARY KEY (`id`),

[0103] UNIQUE key `uidx_uid` (`uid`),

[0104] KEY `inx_create_time` (`create_time`)

[0105]) ENGINE=InnoDBDEFAULT CHARSET=utf8mb4 ROW_FORMAT=DYNAMIC;

[0106] 在将副本数据插入镜像库之后,需要可靠的机制验证数据是否已经被成功存储,可以通过检查插入操作的返回状态确认影响的行数,或者执行一个针对性的查询验证数据确实存在于镜像库中。在确认副本数据成功存储至镜像库后,执行插入操作,插入事务唯一标识符和当前时间戳到主数据库中的第一确认表。副本数据成功存储至镜像库后立即记录确认信息,有效地维护了数据一致性,确保了数据在主数据库和镜像库之间保持同步,降低了数据不一致的风险;同时,通过在第一确认表中插入事务唯一标识符和时间戳,提供了可审计的数据变更历史,为后续的监控提供了可靠的数据。

[0107] 图3示出根据本公开实施例的事务提交的流程示意图,如图3,在副本数据成功存储至镜像库的情况下,执行插入确认记录至第一确认表的操作,确保数据变更的追踪和记录。在副本数据未成功存储至镜像库的情况下,事务回滚至初始状态。若副本数据未能成功存储至镜像库,为了保持数据一致性和完整性,将执行事务回滚,将数据库状态恢复到事务开始前的初始状态,从而防止不完整或错误数据被记录,确保数据的完整性和可靠性。

[0108] 在确认记录成功插入主数据库中创建的第一确认表的情况下,提交事务。若确认记录成功插入,表明所有相关操作都已完成,且数据状态一致。提交事务涉及调用数据库的提交命令,确保所有相关的数据变更都被永久保存。事务的提交在确认记录成功插入第一确认表后进行,保证该事务的数据变更在得到确认和记录后,事务才被视为成功结束。

[0109] 在确认记录未成功插入主数据库中创建的第一确认表的情况下,事务回滚至初始状态。若确认记录未能成功插入,可能由于各种原因,例如数据库连接问题、权限问题或其他异常。在这种情况下,将执行事务回滚,取消自事务开始以来进行的所有数据变更,防止可能的数据不一致或错误,维护了数据的准确性和可靠性。

[0110] S103、在发生主从切换的情况下,根据所述镜像库和从数据库中的第二确认表,获取差异数据。

[0111] 图4示出根据本公开实施例的获取差异数据的流程示意图,如图4,获取差异数据包括:

[0112] S401、对比镜像库和第二确认表,确定所述第二确认表中未记录的事务唯一标识符;第二确认表是基于主从复制机制创建在从数据库中的表。

[0113] 在主从数据库架构中,第一确认表是在主数据库中创建和维护的表,其结构和数据基于数据库的主从复制机制,复制至从数据库中,在从数据库中称为第二确认表。第二确认表在从数据库中维护了主数据库中事务的确认记录的副本,用于在主从切换时或其他特定情况下参与数据一致性的校验。

[0114] 当主从切换发生时,例如主数据库因故障或其他原因而离线时,从数据库将被提升为新的主数据库。由于主从复制机制通常并不保证强一致性,特别是在网络延迟或故障的情况下,从数据库可能未能及时同步数据的最新状态,导致其反映的信息滞后于主数据库。尽管第二确认表在从数据库中提供了事务的确认记录,但其所包含的信息并不代表主数据库中最终的数据变更。通过对比镜像库中数据和第二确认表中的记录,识别和获取由于复制延迟,而未在从数据库中处理的事务对应的事务唯一标识符。

[0115] 获取未在从数据库中处理的事务对应的事务唯一标识符,具体操作包括:

[0116] 从镜像库中提取所有核心数据对应的事务唯一标识符;

[0117] 从第二确认表中提取所有确认记录中的事务唯一标识符;

[0118] 比较上述事务唯一标识符的两个集合,找出在第二确认表中未出现的事务唯一标识符。本公开并不对比较方式进行限定,需要根据具体情况选择合适的比较方式。在一个实施例中,采用在应用层比较的方式,具体步骤包括在数据库中检索数据,并将这些数据作为集合或列表等数据结构存储在应用程序内存中;然后利用应用程序的逻辑进行处理,比如使用循环、条件语句或集合操作比较两个集合,并找出不匹配的事务唯一标识符。该方法提供了高度的灵活性和可适应性,尤其适用于处理逻辑较为复杂的情况。在另一个实施例中,采用在数据库层比较的方式,使用SQL或数据库特定的查询语言执行集合操作,例如使用NOT IN、EXCEPT或LEFT JOIN等语句,直接在数据库中识别出一个集合中存在而另一个集合中不存在的事务唯一标识符。该方法一般更高效,尤其适用于处理大量数据,减少了网络传输和应用服务器的负载。

[0119] S402、基于所述未记录的事务唯一标识符,获取差异数据,所述差异数据为所述镜像库中,所述未记录的事务唯一标识符对应的核心数据。

[0120] 在一个实施例中,获取差异数据的具体实现方式包括根据未记录的事务唯一标识符构造一条或多条查询语句,用于从镜像库中检索对应的核心数据;在数据库中执行上述查询语句,检索每个未记录事务唯一标识符对应的核心数据。

[0121] 主从切换时通过比对镜像库和从数据库中的第二确认表获取差异数据,能够确保

数据在主从数据库间的一致性;通过事务唯一标识符直接定位和获取未在从数据库中同步的事务对应的核心数据,减少了不必要的数据处理和资源消耗。

[0122] S104、基于所述差异数据,构建包括所述差异数据的黑名单。

[0123] 在一个实施例中,构建黑名单的具体实现方式包括定义黑名单的数据结构,数据结构可以是一个列表或集合形式,其中每个元素代表一个需要阻断的数据项。具体地,每个数据项包括表名、字段名和字段名对应的值,黑名单中存储的数据格式为[{表名:{字段名:[字段值...]}},...]。根据上述获取的差异数据填充黑名单,具体实现方式可以是遍历所有由未记录的事务唯一标识符代表的差异数据,并将上述差异数据添加到黑名单中。伪代码如下:

[0124] blacklist = new List()

[0125] for each identifier in difference_identifiers:

[0126] // 假设difference_identifiers是所有未记录的事务唯一标识符

[0127] data = retrieveData(identifier)// 从镜像库检索核心数据

[0128] blacklist.add(data)// 将核心数据添加到黑名单中

[0129] S105:所述黑名单构建完成后,将所述主数据库作为旧主库,将所述从数据库作为新主库,阻断对所述黑名单中记录的所述差异数据的操作。

[0130] 黑名单构建完成后,将当前的主数据库作为旧主库,并将选定的从数据库提升为新主库,通常是通过自动化脚本执行的操作,数据不再写入旧主库,并开始在新主库上接受读写操作。在主从切换后将从数据库提升为新主库,减少业务中断时间,提升业务的连续性和用户的体验感。

[0131] 在一个实施例中,持续拦截对数据库的数据操作语句,对于每个拦截到的数据操作语句,解析并提取出表名、字段名及其相应值等关键组成部分。将提取的数据与黑名单中的记录进行对比,检查是否存在匹配的表名和字段名,以及字段值是否与黑名单的字段值列表相匹配。若发现匹配项,表明当前拦截的数据操作语句试图访问或修改黑名单中记录的某个数据项,立即进行阻断操作。具体的阻断操作步骤包括直接取消当前的数据操作以阻止数据被访问或修改,并记录阻断操作发生的详细信息,如时间、操作类型、涉及的表和字段等,以便于后续的审计和分析。

[0132] 在实施阻断操作时,可以根据系统架构和需求选择在应用层或数据库层实施。在数据库层,触发器或存储过程可以用来在事务执行中检测到黑名单匹配项时,立即抛出异常并取消操作。该方式直接在数据源进行控制,能够有效地阻止不符合数据一致性要求的数据访问或修改行为。在应用层,可以通过在业务逻辑中检测到黑名单匹配项时,中断并取消后续的数据库操作命令,从而阻断对不一致数据的访问或修改操作。该方式提供了更灵活的控制和错误处理。一旦发生阻断,需要对事件进行详细记录。记录到日志系统的信息通常包括时间戳、操作类型、涉及的表和字段名称、拦截的原因等,以便于后续的审计和分析,是未来分析和优化系统行为的重要资源。通常,详细记录的信息会被存储在特定的审计表或日志库中,以确保信息的可靠保存以及方便地回溯分析。

[0133] 在一个实施例中,若某个强一致性要求的表不包含预设规则中定义的核心数据,或者表的数据整体需要被阻断,可以采取对整个表进行拦截的策略。在黑名单中存储如{"表名":{"###ALL###":["1"]}}格式的数据,其"1"是为了满足格式要求而设置的无实际意

义值。当拦截到任何试图对标记为全表拦截的表进行访问或修改操作的语句时,将立即执行阻断操作。全表拦截策略是一种强力的措施,用于处理需要全面保护的表,以确保数据的一致性和安全性。

[0134] 构建黑名单并在新主库生效后阻断对黑名单中数据的操作,能够有效地避免对不一致数据的访问和修改,降低因数据不一致带来的业务风险,提高数据操作的安全性。

[0135] 旧主库恢复后,补齐所述旧主库和所述新主库缺失的数据;释放所述黑名单中的数据。

[0136] 在一个实施例中,旧主库恢复后,数据库管理员在旧主库和新主库上获取全局事务标识符(Global Transaction Identifiers,GTID)集合。GTID是在数据库系统中用来唯一标识每个事务的标识符。当提交事务时,数据库会自动生成一个唯一的GTID。通过比对旧主库和新主库的GTID集合,数据库管理员能够准确地识别出两者之间的事务差异,包括旧主库独有和新主库独有的事务。

[0137] 确定需要补齐的GTID后,数据库管理员从相应数据库的日志系统中提取对应的二进制日志,进行回放操作。二进制日志详细记录了数据库中所有更改操作的序列,并按照时间顺序排列。对于新主库独有的事务,数据库管理员将上述新主库独有的事务对应的二进制日志应用于旧主库,以确保旧主库包含故障期间在新主库上发生的所有数据更改。对于旧主库独有的事务,即宕机前仅在旧主库上执行并完成,而未及时同步到其他从数据库的事务,数据库管理员将上述旧主库独有的事务对应的二进制日志应用于新主库,从而确保新主库能够全面且准确地反映旧主库中发生的所有数据更新。

[0138] 旧主库恢复后,通过同步旧主库与新主库之间的差异数据,有效地保障数据在分布式存储系统中的一致性与完整性,降低了数据冲突和不一致的风险,增强了系统的稳定性和可靠性。同时,为保持业务的连续性和数据的准确性提供了坚实的基础。

[0139] 在完成回放操作后,进行数据一致性检查,确认数据在主数据库和从数据库中的一致性。完成数据一致性检查并确认无误后,释放黑名单中的数据。本公开并不限定具体的释放方式,在一个实施例中,黑名单以数据库形式存储,可以通过执行删除(DELETE)操作移除所有记录,从而清空黑名单。在另一个实施例中,黑名单在应用层构建,采用内存数据结构形式,例如列表或集合,可以通过将黑名单变量重新初始化或设置为空集合的方式进行清空,有效地释放所有被限制的数据项。通过释放黑名单中的数据,能够恢复因一致性问题而暂时被拦截的操作,从而确保数据访问和处理的连续性。不仅提高了系统的响应性和可用性,而且保障了业务流程的正常运行,增强整个系统的稳定性和可靠性。

[0140] 图5示出根据本公开实施例的一种主从数据库的强一致装置的模块示意图。如图5,主从数据库的强一致装置包括:

[0141] 存储模块,用于存储副本数据至镜像库;

[0142] 插入模块,用于所述副本数据成功存储至所述镜像库后,插入确认记录至主数据库中创建的第一确认表;

[0143] 获取模块,用于在发生主从切换的情况下,根据从数据库中的第二确认表和所述镜像库,获取差异数据,所述第二确认表是基于主从复制机制创建在所述从数据库中的表;

[0144] 构建模块,用于基于所述差异数据,构建包括所述差异数据的黑名单;

[0145] 阻断模块,用于所述黑名单构建完成后,将所述主数据库作为旧主库,将所述从数

据库作为新主库,阻断对所述黑名单中记录的所述差异数据的操作。

[0146] 在一些实施例中,本公开实施例提供的装置具有的功能或包含的模块可以用于执行上文方法实施例描述的方法,其具体实现可以参照上文方法实施例的描述,为了简洁,这里不再赘述。

[0147] 本公开实施例还提出一种计算机可读存储介质,上述计算机可读存储介质中存储有至少一条指令或至少一段程序,上述至少一条指令或至少一段程序由处理器加载并执行时实现上述方法。计算机可读存储介质可以是非易失性计算机可读存储介质。

[0148] 本公开实施例还提出一种电子设备,包括:处理器;用于存储处理器可执行指令的存储器;其中,上述处理器被配置为上述方法。

[0149] 电子设备可以被提供为终端、服务器或其它形态的设备。

[0150] 图6示出根据本公开实施例的一种电子设备的框图。例如,电子设备1900可以被提供为一服务器。参照图6,电子设备1900包括处理组件1922,其进一步包括一个或多个处理器,以及由存储器1932所代表的存储器资源,用于存储可由处理组件1922的执行的指令,例如应用程序。存储器1932中存储的应用程序可以包括一个或一个以上的每一个对应于一组指令的模块。此外,处理组件1922被配置为执行指令,以执行上述方法。

[0151] 电子设备1900还可以包括一个电源组件1926被配置为执行电子设备1900的电源管理,一个有线或无线网络接口1950被配置为将电子设备1900连接到网络,和一个输入输出(I/O)接口1958。电子设备1900可以操作基于存储在存储器1932的操作系统,例如Windows ServerTM,Mac OS XTM,UnixTM,LinuxTM,FreeBSDTM或类似。

[0152] 这里参照根据本公开实施例的方法、装置(系统)的流程图和/或框图描述了本公开的各个方面。应当理解,流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合,都可以由计算机可读程序指令实现。

[0153] 这些计算机可读程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器,从而生产出一种机器,使得这些指令在通过计算机或其它可编程数据处理装置的处理器执行时,产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中,这些指令使得计算机、可编程数据处理装置和/或其他设备以特定方式工作,从而,存储有指令的计算机可读介质则包括一个制造品,其包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的各个方面的指令。

[0154] 也可以把计算机可读程序指令加载到计算机、其它可编程数据处理装置、或其它设备上,使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤,以产生计算机实现的过程,从而使得在计算机、其它可编程数据处理装置、或其它设备上执行的指令实现流程图和/或框图中的一个或多个方框中规定的功能/动作。

[0155] 附图中的流程图和框图显示了根据本公开的多个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或指令的一部分,上述模块、程序段或指令的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。在有些作为替换的实现中,方框中所标准的功能也可以以不同于附图中所标准的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的是,框图和/或

流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或动作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0156] 以上已经描述了本公开的各实施例，上述说明是示例性的，并非穷尽性的，并且也不限于所披露的各实施例。在不偏离所说明的各实施例的范围和精神的情况下，对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择，旨在最好地解释各实施例的原理、实际应用或对市场中的技术改进，或者使本技术领域的其它普通技术人员能理解本文披露的各实施例。

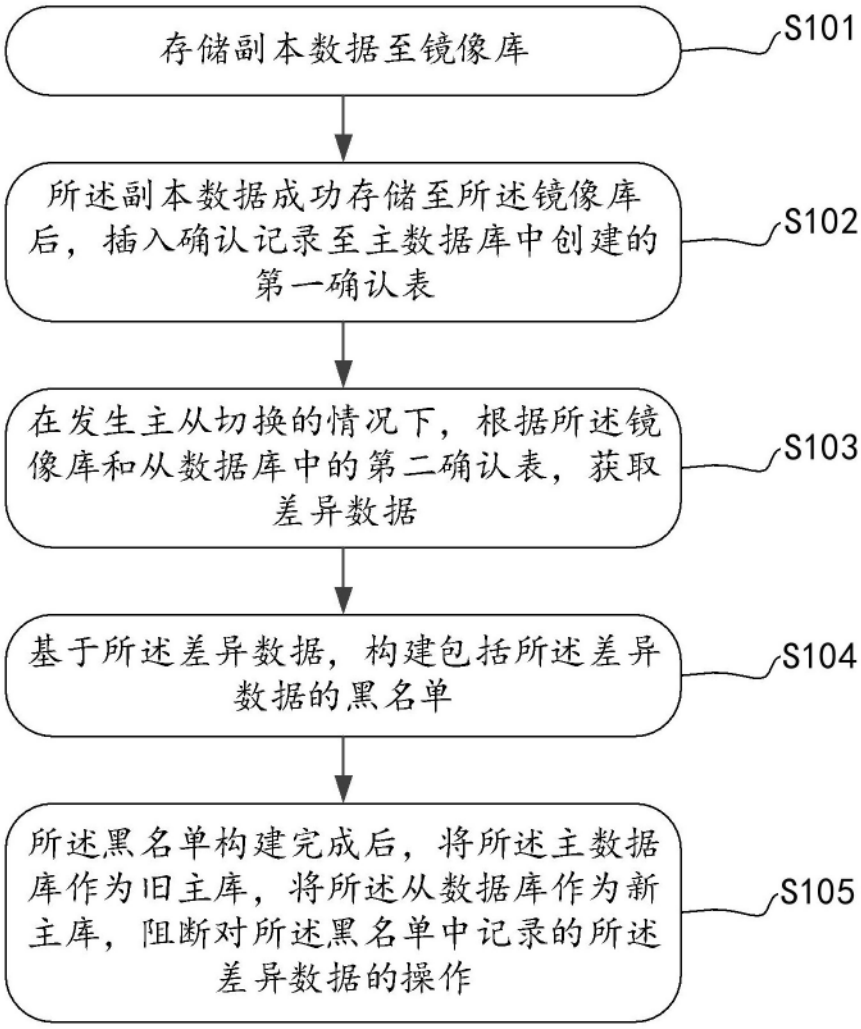

存储副本数据至镜像库
S101
所述副本数据成功存储至所述镜像库后，插入确认记录至主数据库中创建的第一确认表
S102
在发生主从切换的情况下，根据所述镜像库和从数据库中的第二确认表，获取差异数据
S103
基于所述差异数据，构建包括所述差异数据的黑名单
S104
所述黑名单构建完成后，将所述主数据库作为旧主库，将所述从数据库作为新主库，阻断对所述黑名单中记录的所述差异数据的操作
S105

图1

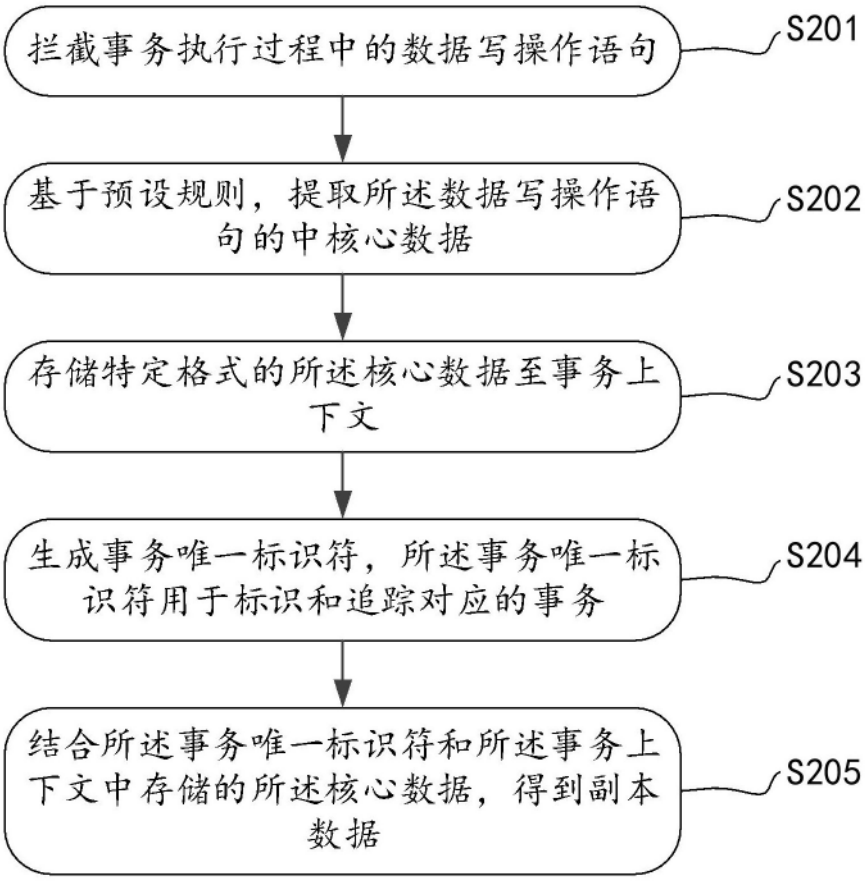

拦截事务执行过程中的数据写操作语句
S201
基于预设规则，提取所述数据写操作语句的中核心数据
S202
存储特定格式的所述核心数据至事务上下文
S203
生成事务唯一标识符，所述事务唯一标识符用于标识和追踪对应的事务
S204
结合所述事务唯一标识符和所述事务上下文中存储的所述核心数据，得到副本数据
S205

图2

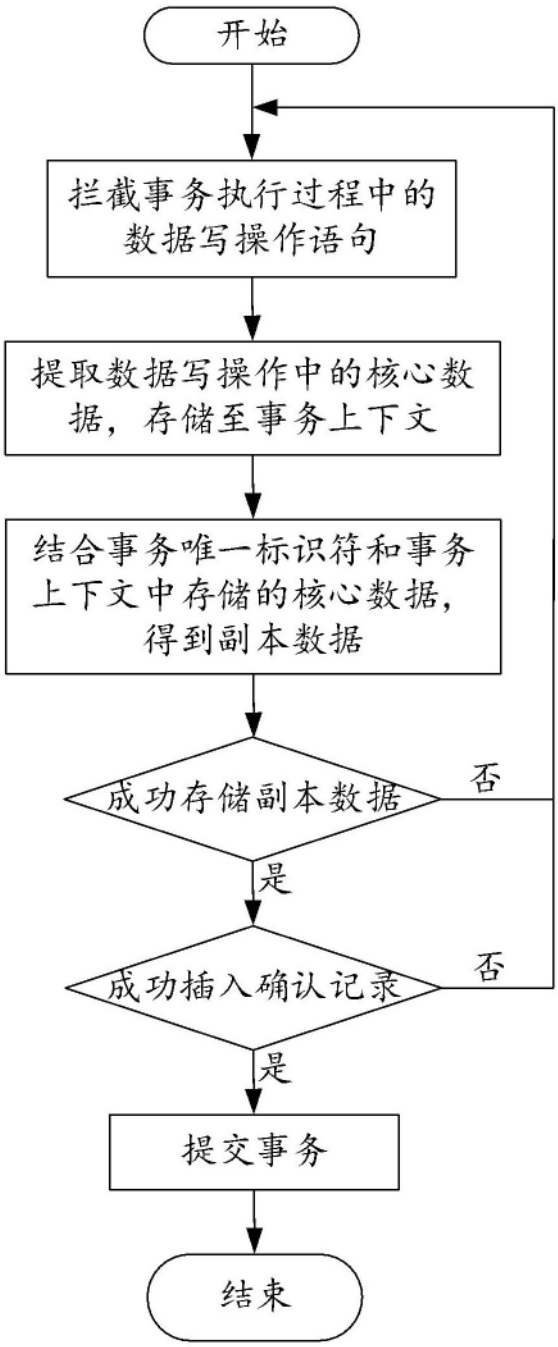

开始
拦截事务执行过程中的数据写操作语句
提取数据写操作中的核心数据，存储至事务上下文
结合事务唯一标识符和事务上下文中存储的核心数据，得到副本数据
成功存储副本数据
否
是
成功插入确认记录
否
是
提交事务
结束

图3

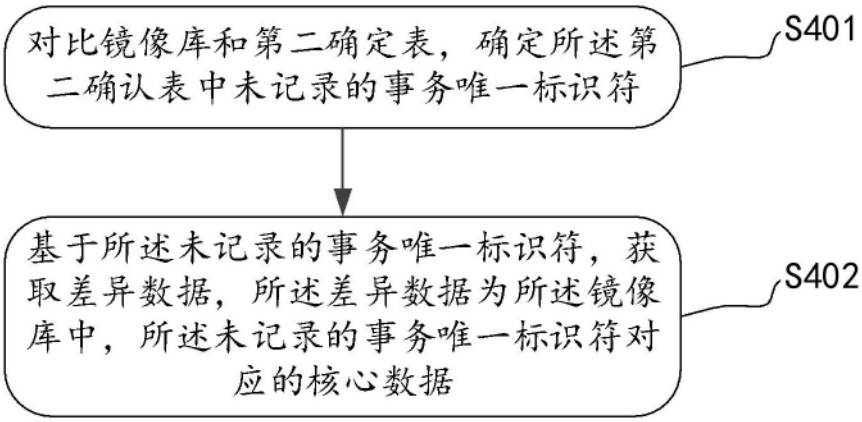

对比镜像库和第二确定表，确定所述第二确认表中未记录的事务唯一标识符
S401
基于所述未记录的事务唯一标识符，获取差异数据，所述差异数据为所述镜像库中，所述未记录的事务唯一标识符对应的核心数据
S402

图4

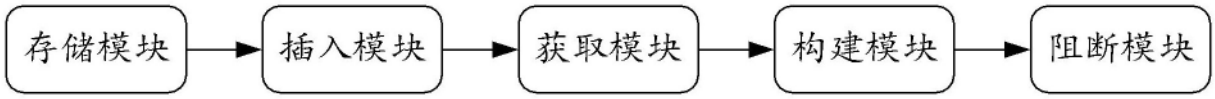

存储模块
插入模块
获取模块
构建模块
阻断模块

图5

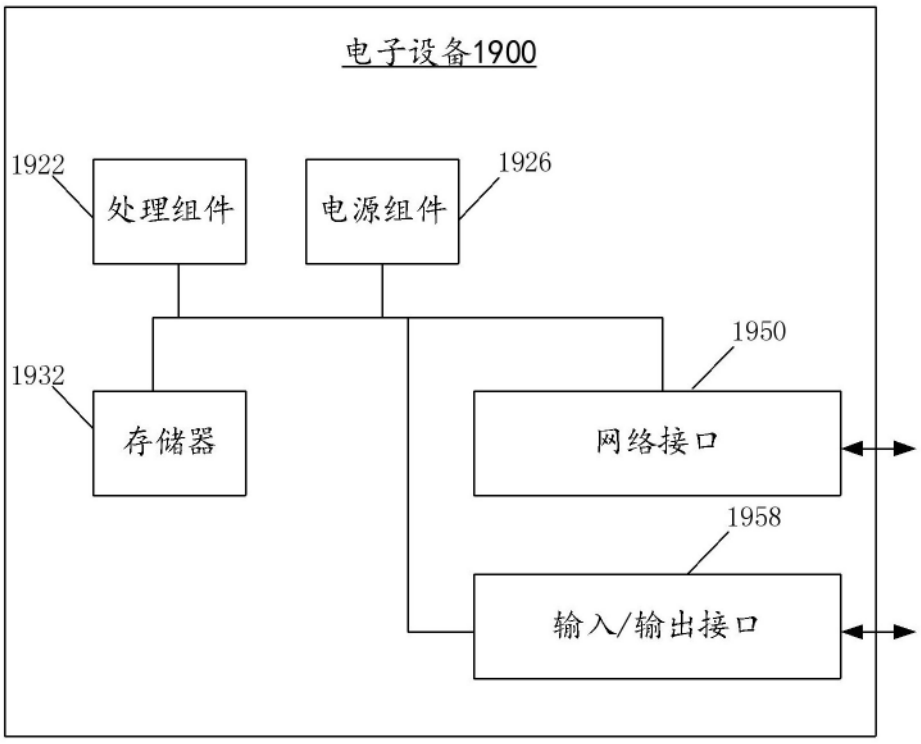

电子设备1900
1922
处理组件
1926
电源组件
1932
存储器
1950
网络接口
1958
输入/输出接口

图6