



## (12)发明专利

(10)授权公告号 CN 107229455 B

(45)授权公告日 2019.09.17

(21)申请号 201610173747.6

(22)申请日 2016.03.24

(65)同一申请的已公布的文献号

申请公布号 CN 107229455 A

(43)申请公布日 2017.10.03

(73)专利权人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼资本大厦一座四  
层847号邮箱

(72)发明人 杜娟

(74)专利代理机构 北京博思佳知识产权代理有  
限公司 11415

代理人 林祥

(51)Int.Cl.

G06F 9/448(2018.01)

(56)对比文件

CN 104253702 A,2014.12.31,

CN 103853764 A,2014.06.11,

CN 102035865 A,2011.04.27,

US 8751598 B1,2014.06.10,

CN 103716253 A,2014.04.09,

审查员 张静

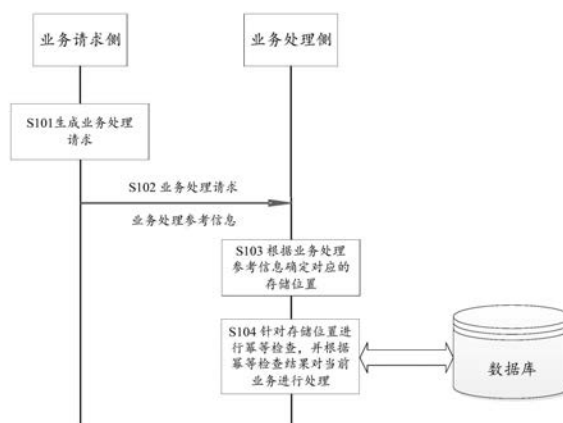
权利要求书3页 说明书8页 附图3页

(54)发明名称

一种业务处理方法、装置及系统

(57)摘要

本申请公开了一种业务处理方法、装置及系统。一种业务处理方法包括：业务请求侧生成携带业务处理参考信息的业务处理请求，将所述业务处理请求发送至业务处理侧；针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息；业务处理侧接收到所述业务处理请求后，根据请求中携带的业务处理参考信息确定对应的存储位置，针对所确定的存储位置进行幂等检查，根据幂等检查结果对当前业务进行处理。应用本申请技术方案，不会出现针对同一业务事件的多次幂等检查在不同存储位置进行的情况，从而有效地避免了业务重复处理的问题。



1. 一种业务处理方法,其特征在于,该方法包括:

业务请求侧生成携带业务处理参考信息的业务处理请求,将所述业务处理请求发送至业务处理侧;所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置,针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息;

业务处理侧接收到所述业务处理请求后,根据请求中携带的业务处理参考信息确定对应的存储位置,针对所确定的存储位置进行幂等检查,根据幂等检查结果对当前业务进行处理。

2. 根据权利要求1所述的方法,其特征在于,所述业务处理参考信息为时刻信息;

所述生成携带业务处理参考信息的业务处理请求,包括:针对同一业务事件,如果首次发送业务处理请求,则获取当前的时刻信息,将该时刻信息写入业务处理请求中,后续针对该业务事件重发的请求中均携带该时刻信息;

所述根据请求中携带的业务处理参考信息确定对应的存储位置,包括:根据请求中携带的时刻信息,确定与该时刻信息对应的时间维度数据分表。

3. 根据权利要求1所述的方法,其特征在于,所述业务处理参考信息为数据库运行模式信息,所述数据库模式包括:正常模式、故障切换模式;

所述生成携带业务处理参考信息的业务处理请求,包括:针对同一业务事件,如果首次发送业务处理请求,则获取业务处理侧当前所使用的数据库运行模式信息,将该数据库运行模式信息写入业务处理请求中,后续针对该业务事件重发的请求中均携带该数据库运行模式信息;

所述根据请求中携带的业务处理参考信息确定对应的存储位置,包括:根据请求中携带的数据库运行模式信息,确定与该数据库运行模式信息对应的数据库。

4. 一种业务处理方法,应用于业务请求侧,其特征在于,该方法包括:

生成携带业务处理参考信息的业务处理请求;

将所述业务处理请求发送至业务处理侧,以使得业务处理侧接收到所述业务处理请求后,根据请求中携带的业务处理参考信息确定对应的存储位置,针对所确定的存储位置进行幂等检查,根据幂等检查结果对当前业务进行处理;

其中,所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置,针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息。

5. 根据权利要求4所述的方法,其特征在于,所述业务处理参考信息为时刻信息;

所述生成携带业务处理参考信息的业务处理请求,包括:针对同一业务事件,如果首次发送业务处理请求,则获取当前的时刻信息,将该时刻信息写入业务处理请求中,后续针对该业务事件重发的请求中均携带该时刻信息。

6. 根据权利要求4所述的方法,其特征在于,所述业务处理参考信息为数据库运行模式信息,所述数据库模式包括:正常模式、故障切换模式;

所述生成携带业务处理参考信息的业务处理请求,包括:针对同一业务事件,如果首次发送业务处理请求,则获取业务处理侧当前所使用的数据库运行模式信息,将该数据库运行模式信息写入业务处理请求中,后续针对该业务事件重发的请求中均携带该数据库运行模式信息。

7. 一种业务处理方法,应用于业务处理侧,其特征在于,该方法包括:

接收业务请求侧发送的业务处理请求,所述业务处理请求中携带业务处理参考信息;

根据请求中携带的业务处理参考信息确定对应的存储位置,针对所确定的存储位置进行幂等检查,根据幂等检查结果对当前业务进行处理;

其中,所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置,业务请求侧针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息。

8. 根据权利要求7所述的方法,其特征在于,所述业务处理参考信息为时刻信息;

所述根据请求中携带的业务处理参考信息确定对应的存储位置,包括:根据请求中携带的时刻信息,确定与该时刻信息对应的时间维度数据分表。

9. 根据权利要求7所述的方法,其特征在于,所述业务处理参考信息为数据库运行模式信息,所述数据库模式包括:正常模式、故障切换模式;

所述根据请求中携带的业务处理参考信息确定对应的存储位置,包括:根据请求中携带的数据库运行模式信息,确定与该数据库运行模式信息对应的数据库。

10. 一种业务处理装置,应用于业务请求侧,其特征在于,该装置包括:

请求生成模块,用于生成携带业务处理参考信息的业务处理请求;

请求发送模块,用于将所述业务处理请求发送至业务处理侧,以使得业务处理侧接收到所述业务处理请求后,根据请求中携带的业务处理参考信息确定对应的存储位置,针对所确定的存储位置进行幂等检查,根据幂等检查结果对当前业务进行处理;

其中,所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置,针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息。

11. 根据权利要求10所述的装置,其特征在于,所述业务处理参考信息为时刻信息;

所述请求生成模块,具体用于:针对同一业务事件,如果首次发送业务处理请求,则获取当前的时刻信息,将该时刻信息写入业务处理请求中,后续针对该业务事件重发的请求中均携带该时刻信息。

12. 根据权利要求10所述的装置,其特征在于,所述业务处理参考信息为数据库运行模式信息,所述数据库模式包括:正常模式、故障切换模式;

所述请求生成模块,具体用于:针对同一业务事件,如果首次发送业务处理请求,则获取业务处理侧当前所使用的数据库运行模式信息,将该数据库运行模式信息写入业务处理请求中,后续针对该业务事件重发的请求中均携带该数据库运行模式信息。

13. 一种业务处理装置,应用于业务处理侧,其特征在于,该装置包括:

请求接收模块,用于接收业务请求侧发送的业务处理请求,所述业务处理请求中携带业务处理参考信息;

业务处理模块,用于根据请求中携带的业务处理参考信息确定对应的存储位置,针对所确定的存储位置进行幂等检查,根据幂等检查结果对当前业务进行处理;

其中,所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置,业务请求侧针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息。

14. 根据权利要求13所述的装置,其特征在于,所述业务处理参考信息为时刻信息;

所述业务处理模块,具体用于:根据请求中携带的时刻信息,确定与该时刻信息对应的时间维度数据分表,针对所确定的数据分表进行幂等检查,根据幂等检查结果对当前业务进行处理。

15.根据权利要求13所述的装置,其特征在于,所述业务处理参考信息为数据库运行模式信息,所述数据库模式包括:正常模式、故障切换模式;

所述业务处理模块,具体用于:根据请求中携带的数据库运行模式信息,确定与该数据库运行模式信息对应的数据库,针对所确定的数据库进行幂等检查,根据幂等检查结果对当前业务进行处理。

16.一种业务处理系统,其特征在于,该系统包括业务请求侧装置及业务处理侧装置:

所述业务请求侧装置包括:

请求生成模块,用于生成携带业务处理参考信息的业务处理请求;

请求发送模块,用于将所述业务处理请求发送至业务处理侧;

其中,所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置,针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息;

所述业务处理侧装置包括:

请求接收模块,用于接收业务请求侧发送的业务处理请求;

业务处理模块,用于根据请求中携带的业务处理参考信息确定对应的存储位置,针对所确定的存储位置进行幂等检查,根据幂等检查结果对当前业务进行处理。

## 一种业务处理方法、装置及系统

### 技术领域

[0001] 本申请涉及计算机应用技术领域,尤其涉及一种业务处理方法、装置及系统。

### 背景技术

[0002] 在计算机系统中,幂等操作的特点是其任意多次执行所产生的影响均与一次执行的所产生的影响相同,在实际应用中,幂等操作常用于避免重复操作对系统的造成的影响。以金融系统为例,针对一次账务转入或转出请求,必须保证该请求只导致一次实际的转账处理,不允许由于网络延迟等原因导致多次转账处理,因此金融系统都需要保证其对外账务接口的幂等性,。

[0003] 为了实现幂等性,现有技术的常用做法是在业务系统侧的数据库中针对每次业务请求建立唯一性约束,例如:在数据库中建立幂等表,每次处理业务请求之前,均要求将该业务请求的某种唯一标识信息(例如“业务请求单号”、“业务请求单号+业务类型”、“业务请求单号+业务请求来源”等)插入这张表,如果插入成功,说明业务系统是第一次处理该业务请求,则继续正常的业务处理流程;反之如果插入失败,则说明幂等表中已经存在该业务的信息,即业务系统之前曾经处理过该业务请求,那么可以直接确定该请求为重复请求,进而停止处理该请求。

[0004] 在一般情况下,上述方案可以较好地实现业务系统对外接口的幂等性。然而基于分布式存储或灾备等需求,对数据进行分表或分库存储的方案应用越来越普遍,而在分表或分库的应用场景下,可能会出现的一种情况是:针对同一业务,请求方向业务系统发起了多次请求,而出于某种原因(例如灾备切换等),多次请求被业务系统路由到了不同的数据表,由于不同数据表所使用的幂等表不同,因此很可能出现幂等失效,进而导致业务重复处理。

### 发明内容

[0005] 针对上述技术问题,本申请提供一种业务处理方法、装置及系统,以解决在分库或多库应用场景下的幂等失效问题,技术方案如下:

[0006] 根据本申请的第一方面,提供一种业务处理方法,该方法包括:

[0007] 业务请求侧生成携带业务处理参考信息的业务处理请求,将所述业务处理请求发送至业务处理侧;所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置,针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息;

[0008] 业务处理侧接收到所述业务处理请求后,根据请求中携带的业务处理参考信息确定对应的存储位置,针对所确定的存储位置进行幂等检查,根据幂等检查结果对当前业务进行处理。

[0009] 根据本申请的第二方面,提供一种业务处理方法,应用于业务请求侧,该方法包括:

[0010] 生成携带业务处理参考信息的业务处理请求；

[0011] 将所述业务处理请求发送至业务处理侧，以使得业务处理侧接收到所述业务处理请求后，根据请求中携带的业务处理参考信息确定对应的存储位置，针对所确定的存储位置进行幂等检查，根据幂等检查结果对当前业务进行处理；

[0012] 其中，所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置，针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息。

[0013] 根据本申请的第三方面，提供一种业务处理方法，应用于业务处理侧，该方法包括：

[0014] 接收业务请求侧发送的业务处理请求，所述业务处理请求中携带业务处理参考信息；

[0015] 根据请求中携带的业务处理参考信息确定对应的存储位置，针对所确定的存储位置进行幂等检查，根据幂等检查结果对当前业务进行处理；

[0016] 其中，所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置，业务请求侧针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息。

[0017] 根据本申请的第四方面，提供一种业务处理装置，应用于业务请求侧，该装置包括：

[0018] 请求生成模块，用于生成携带业务处理参考信息的业务处理请求；

[0019] 请求发送模块，用于将所述业务处理请求发送至业务处理侧，以使得业务处理侧接收到所述业务处理请求后，根据请求中携带的业务处理参考信息确定对应的存储位置，针对所确定的存储位置进行幂等检查，根据幂等检查结果对当前业务进行处理；

[0020] 其中，所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置，针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息。

[0021] 根据本申请的第五方面，提供一种业务处理装置，应用于业务处理侧，该装置包括：

[0022] 请求接收模块，用于接收业务请求侧发送的业务处理请求，所述业务处理请求中携带业务处理参考信息；

[0023] 业务处理模块，用于根据请求中携带的业务处理参考信息确定对应的存储位置，针对所确定的存储位置进行幂等检查，根据幂等检查结果对当前业务进行处理；

[0024] 其中，所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置，业务请求侧针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息。

[0025] 根据本申请的第六方面，提供一种业务处理系统，该系统包括业务请求侧装置及业务处理侧装置：

[0026] 所述业务请求侧装置包括：

[0027] 请求生成模块，用于生成携带业务处理参考信息的业务处理请求；

[0028] 请求发送模块，用于将所述业务处理请求发送至业务处理侧；

[0029] 其中,所述业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置,针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息;

[0030] 所述业务处理侧装置包括:

[0031] 请求接收模块,用于接收业务请求侧发送的业务处理请求;

[0032] 业务处理模块,用于根据请求中携带的业务处理参考信息确定对应的存储位置,针对所确定的存储位置进行幂等检查,根据幂等检查结果对当前业务进行处理。

[0033] 本申请所提供的技术方案,在业务请求侧向业务处理侧发送的业务处理请求中添加了业务处理参考信息,由于针对同一业务事件多次发送的业务处理请求中所携带的业务处理参考信息都是相同的,因此不会出现针对同一业务事件的多次幂等检查在不同存储位置进行的情况,从而有效地避免了业务重复处理的问题。

[0034] 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,并不能限制本申请。

## 附图说明

[0035] 为了更清楚地说明本申请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,下面描述中的附图仅仅是本申请中记载的一些实施例,对于本领域普通技术人员来讲,还可以根据这些附图获得其他的附图。

[0036] 图1是本申请的业务处理方法的流程示意图;

[0037] 图2是本申请的数据库部署方案示意图;

[0038] 图3是本申请的业务请求侧装置的结构示意图;

[0039] 图4是本申请的业务处理侧装置的结构示意图;

[0040] 图5是本申请的业务处理系统的结构示意图。

## 具体实施方式

[0041] 在海量数据的应用场景下,为了实现数据的分布式存储,一般会对采用分表的方式对数据进行存储,分表可以基于多种维度实现,其中一种常用的划分维度是基于“时间”维度进行分表,例如按照1年12个月划分出12个分表,分别用于存储在不同时间发起的业务请求所对应的业务数据。

[0042] 此外,为了保证系统的容灾特性,对数据库进行冗余备份也是一种常见的处理方案,即将主库中的数据完全复制到F0 (Failover,故障切换) 库中,主库和F0库一般被部署在不同的区域,正常状态下,业务请求被路由到主库(正常模式),灾备状态下业务请求将被切换路由至F0库(故障切换模式)。

[0043] 分库和分表都是数据库应用的常见方案,然而在分库和分表的应用场景下,可能会导致现有的幂等性实现方案失效,场景举例如下:

[0044] 假设用户A希望向用户B转账100元,业务处理侧的转账平台在11月30日23时59分接收到A通过某应用平台发送的转账请求,将该业务路由到分表11(对11月对应分表的统称),分表11的幂等表中可以成功插入该业务的标识信息,因此正常处理该业务:将100元从

账户A转到账户B。但是由于网络问题,业务请求侧的应用平台并没有收到业务已成功处理的响应,于是判定为处理超时并重新向转账平台发送第二次转账请求。转账平台收到第二次转账请求的时间为12月1日00时01分,将该业务路由到分表12(对12月对应分表的统称),而分表12的幂等表中并不存在该转账业务的标识信息,因此同样可以成功插入并且正常处理业务,导致A再一次向B转账100元。

[0045] 假设用户A希望向用户B转账100元,业务处理侧的转账平台第一次接收到A通过某应用平台发送的转账请求时,系统处于正常运行状态,因此将该业务路由到主库中的数据表,主库的幂等表中可以成功插入该业务的标识信息,因此正常处理该业务:将100元从账户A转到账户B;由于网络问题,业务请求侧的应用平台并没有收到业务已成功处理的响应,于是判定为处理超时并重新向转账平台发送第二次转账请求。转账平台收到第二次转账请求时,主库出现故障,因此将该业务路由到F0库中的数据表,而F0库的幂等表中并不存在该转账业务的标识信息,因此同样可以成功插入并且正常处理业务,导致A再一次向B转账100元。

[0046] 可见,在数据分库或多库存储的应用场景下,由于不同的存储位置(例如主库/F0库、分库、分表等等)分别维护各自的幂等表,因此当同一业务事件的多次请求被路由到不同的存储位置时,就会导致幂等检查失效的问题。

[0047] 针对上述问题,本申请提供一种业务处理方法,参见图1所示,该方法可以包括以下步骤:

[0048] S101,业务请求侧生成携带业务处理参考信息的业务处理请求;

[0049] 与现有技术方案相比,本申请方案在业务处理请求中添加扩展参数:业务处理参考信息,该信息的一个作用是在分库或分表的应用场景下,令业务处理侧能够根据该信息确定出处理当前业务所使用的数据库存储位置,即在处理当前业务的过程中,相关的数据处理操作需要在哪个数据库(包括主库/F0、分库等)和/或哪张数据表中进行,以便业务处理侧在所确定的存储位置中进行后续的数据处理操作。

[0050] 本申请方案并不要求业务请求了解业务处理侧的实际分库或分表逻辑,只需业务请求侧所提供的业务处理参考信息能够被业务处理侧识别、并且根据该信息确定处理当前业务应该使用的具体数据库或数据表即可。

[0051] 例如:对于数据库按“时间”维度进行分表的情况,业务请求侧只需在发送业务处理请求前,获取当前的时刻信息,并且将该时刻信息写入业务处理请求中。进而业务处理侧可以根据该时刻信息,选择对应的时间维度数据分表来进行后续处理。

[0052] 对于数据库冗余备份的应用场景,业务请求侧需要在发送业务处理请求前,感知业务处理侧当前所使用的数据库运行模式信息,这里所说的运行模式包括正常运行模式和故障切换模式,然后将该数据库运行模式信息写入业务处理请求中。进而业务处理侧可以根据该数据库运行模式信息,选择主库或F0库来进行后续的处理。实际应用时,业务请求侧可以以订阅的方式获取业务请求侧或中间件主动推送的数据库运行模式信息,也可以在需要发送业务处理请求时按需获取数据库运行模式信息,本申请并不需要进行限定。

[0053] 需要说明的是,上述两种“业务处理参考信息”仅用于示意性说明,不应理解为对本申请方案的限定,例如,在业务请求侧了解业务处理侧的分库分表逻辑的情况下,也可以直接将处理当前业务所使用的数据库或数据表的标识添加到业务请求中。另外可以理解的



是,根据实际的应用需求,在一条业务处理请求中,允许同时携带多种类型的业务处理参考信息,例如同时携带上述的时刻信息及数据库运行模式信息,以便同时满足按时间维度分表及故障切换的场景应用需求。

[0054] S102,业务请求侧将业务处理请求发送至业务处理侧;

[0055] 本申请方案中,业务处理参考信息的另一个作用是在分库分表的应用场景下,保证业务处理侧接口的幂等性。具体而言,要求业务请求侧针对同一业务事件多次发送的业务处理请求中所携带的业务处理参考信息应该是相同的,这样就能够令业务处理侧在处理基于同一业务事件的多次请求时,都针对相同的存储位置进行幂等检查,从而避免幂等检查失效的问题发生。

[0056] 实际应用时,业务请求侧可以在首次发送业务处理请求时,获取业务处理参考信息(例如当前时刻信息、当前数据库运行模式信息等),将该信息写入业务处理请求中,并且对首次写入的业务处理参考信息以业务事件为标识进行存储。后续如果需要再次发送针对同一业务事件的处理请求,则直接读取之前存储的业务处理参考信息并将其写入新的业务处理请求中。当然,本申请方案只需保证针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息即可,因此上述方式并不应理解为对本申请方案的限定,例如,业务请求侧也可以直接对首次生成的业务处理请求消息体的全部或一部分(其中至少应包含首次使用的业务处理参考信息)进行保存,后续有重发需求时,通过读取预先保存的内容就可以实现重发,并且保证多次请求中携带的业务处理参考信息相同。

[0057] S103,业务处理侧接收到业务处理请求后,根据请求中携带的业务处理参考信息确定对应的存储位置;

[0058] 本申请方案中,业务处理侧需要根据业务请求侧提供的业务处理参考信息来确定:处理当前业务时,相关的数据处理操作需要在哪个数据库(包括主库/F0、分库等)和/或哪张数据表中进行。

[0059] 例如:如果在业务处理请求中携带了时刻信息,则业务处理侧可以根据该时刻信息确定对应的时间维度数据分表;如果在业务处理请求中携带了数据库运行模式信息,则业务处理侧根据该信息判断是使用主库还是F0库。

[0060] 应该理解的是,在实际应用中,上述业务处理参考信息可能仅是用于确定存储位置的一部分信息,例如:在基于“用户编号”和“时间”的二维分表方案中,需要结合“发起当前请求的用户编号(该信息属于业务请求中默认携带的信息)”以及“时刻信息”来确定具体的数据分表;在确定主库或F0库后,还需要根据分表方案来进一步确定数据分表,等等。当然,“其他信息”都属于根据现有技术方案可以正常获取的信息,利用其他信息确定存储位置也可以采用与现有技术一致的方案,这些并不影响本申请方案的实现。

[0061] S104,针对所确定的存储位置进行幂等检查,并根据幂等检查结果对当前业务进行处理;

[0062] 在确定业务相关数据在数据库中存储位置的前提下,本申请方案中的幂等检查可以采用与现有技术相同的方案,即:每次处理业务请求之前,将该业务请求的某种唯一标识信息(例如“业务请求单号”、“业务请求单号+业务类型”、“业务请求单号+业务请求来源”等)插入当前存储位置对应的幂等表,如果插入成功,说明业务系统是第一次处理该业务请求,则继续正常的业务处理流程;反之如果插入失败,则说明幂等表中已经存在该业务的信

息,即业务系统之前曾经处理过该业务请求,那么可以直接确定该请求为重复请求,进而停止处理该请求。当然,本申请并不需要对具体的幂等检查方案进行限定。

[0063] 下面结合将结合具体的应用场景,对本申请的方案进行说明:

[0064] 假设在一个金融系统中,转账平台侧使用的业务数据库部署情况如图2所示,其中数据分表基于两个维度:

[0065] 在“用户编号”维度:取用户编号的其中2位,形成00-99共100个用户分位。

[0066] 在“时间”维度:按照1年12个月形成12个分位。

[0067] 结合上述两种维度,总共形成12\*100共1200张分表,每张分表的命名规则为“用户编号分位\_月份分位”,且每张分表各自对应一张幂等表。

[0068] 此外,为了实现系统容灾,将主库中的数据全量备份到F0库中,F0库与主库的分表结构相同。正常状态下,业务请求被路由到主库(正常模式),灾备状态下业务请求将被切换路由至F0库(故障切换模式)。

[0069] 假设用户A(用户编号为100002)希望向用户B转账100元,通过应用平台向转账平台发起转账请求。应用平台首次发起转账请求的时间为2015年11月30日23时59分,并且此时数据库运行于正常模式,则应用平台将“2015-11-3023:59”以及“正常模式”写入转账请求中。

[0070] 转账平台接收到上述转账请求后,首先根据请求中携带的信息确定存储位置为主库02\_11表,然后执行将该转账业务的流水单号插入主库02\_11表的幂等表的操作,由于插入成功,因此正常处理该业务:将100元从账户A转到账户B。

[0071] 情景1:假设由于网络问题,应用平台并没有收到业务已成功处理的响应,于是判定为处理超时并重新向转账平台发送第二次转账请求。第二次发起转账请求的时间为2015年12月1日00时01分,但是在第二次转账请求中,携带的存储位置信息仍然为“2015-11-30 23:59”以及“正常模式”。

[0072] 转账平台接收到第二次转账请求后,根据请求中携带的信息确定存储位置为主库02\_11表,并进一步执行将该转账业务的流水单号插入主库02\_11表的幂等表的操作,由于幂等表中已经存在相同的记录,因此插入失败,进而停止处理本次转账请求,避免了错误的重复转账处理。

[0073] 情景2:仍然假设由于网络问题,应用平台并没有收到业务已成功处理的响应,于是判定为处理超时并重新向转账平台发送第二次转账请求。第二次发起转账请求的时间为2015年12月1日00时01分,并且此时转账系统侧的数据库已经切换至故障切换模式。但是在第二次转账请求中,携带的信息仍然为“2015-11-30 23:59”以及“正常模式”。

[0074] 转账平台接收到第二次转账请求后,根据请求中携带的信息确定存储位置为主库02\_11表,并进一步执行将该转账业务的流水单号插入主库02\_11表的幂等表的操作,但是此时主库故障无法访问,因此本次业务请求无法处理,结果是同样避免了错误的重复转账处理。

[0075] 可以理解的是,上述金融系统的应用实例仅用于示意性说明,并不构成对本申请方案应用场景的限制。事实上,对于系统数据库涉及数据分库或多库存储、且不同的存储位置(例如主库/F0库、分库、分表等等)分别维护各自幂等表的应用场景,都可以应用本申请方案保证系统接口的幂等性。根据本申请方案,由于业务请求侧针对同一业务事件多次发

送的业务处理请求中所携带的业务处理参考信息都是相同的,因此本申请方案不会出现针对同一业务事件的多次幂等检查在不同存储位置进行的情况,从而有效地避免业务重复处理的问题。

[0076] 相应于上述方法实施例,本申请还提供一种应用于业务请求侧的业务处理装置,参见图3所示,该装置可以包括:

[0077] 请求生成模块110,用于生成携带业务处理参考信息的业务处理请求;

[0078] 请求发送模块120,用于将业务处理请求发送至业务处理侧,以使得业务处理侧接收到业务处理请求后,根据请求中携带的业务处理参考信息确定对应的存储位置,针对所确定的存储位置进行幂等检查,根据幂等检查结果对当前业务进行处理;

[0079] 其中,业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置,针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息。

[0080] 在本申请的一种具体实施方式中,业务处理参考信息可以为时刻信息;

[0081] 请求生成模块110可以具体用于:针对同一业务事件,如果首次发送业务处理请求,则获取当前的时刻信息,将该时刻信息写入业务处理请求中,后续针对该业务事件重发的请求中均携带该时刻信息。

[0082] 在本申请的另一种具体实施方式中,业务处理参考信息可以为数据库运行模式信息,数据库模式可以包括:正常模式、故障切换模式;

[0083] 请求生成模块110可以具体用于:针对同一业务事件,如果首次发送业务处理请求,则获取业务处理侧当前所使用的数据库运行模式信息,将该数据库运行模式信息写入业务处理请求中,后续针对该业务事件重发的请求中均携带该数据库运行模式信息。

[0084] 参见图4所示,本申请还提供一种应用于业务处理侧的业务处理装置,该装置可以包括:

[0085] 请求接收模块210,用于接收业务请求侧发送的业务处理请求,业务处理请求中携带业务处理参考信息;

[0086] 业务处理模块220,用于根据请求中携带的业务处理参考信息确定对应的存储位置,针对所确定的存储位置进行幂等检查,根据幂等检查结果对当前业务进行处理;

[0087] 其中,业务处理参考信息用于业务处理侧确定当前业务的相关数据在数据库中的存储位置,业务请求侧针对同一业务事件多次发送的业务处理请求中携带相同的业务处理参考信息。

[0088] 在本申请的一种具体实施方式中,业务处理参考信息可以为时刻信息;

[0089] 业务处理模块220可以具体用于:根据请求中携带的时刻信息,确定与该时刻信息对应的时间维度数据分表,针对所确定的数据分表进行幂等检查,根据幂等检查结果对当前业务进行处理。

[0090] 在本申请的另一种具体实施方式中,业务处理参考信息可以为数据库运行模式信息,数据库模式可以包括:正常模式、故障切换模式;

[0091] 业务处理模块220可以具体用于:根据请求中携带的数据库运行模式信息,确定与该数据库运行模式信息对应的数据库,针对所确定的数据库进行幂等检查,根据幂等检查结果对当前业务进行处理。

[0092] 本申请还提供一种业务处理系统,如图5所示,该系统可以包括业务请求侧装置

100及业务处理侧装置200,两侧装置的具体实施方式可参见前面的实施例,本申请中不再重复说明。

[0093] 通过以上的实施方式的描述可知,本领域的技术人员可以清楚地了解到本申请可借助软件加必需的通用硬件平台的方式来实现。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品可以存储在存储介质中,如ROM/RAM、磁碟、光盘等,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)执行本申请各个实施例或者实施例的某些部分所述的方法。

[0094] 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,对于装置或系统实施例而言,由于其基本相似于方法实施例,所以描述得比较简单,相关之处参见方法实施例的部分说明即可。以上所描述的装置或系统实施例仅仅是示意性的,其中所述作为分离部件说明的模块可以是或者也可以不是物理上分开的,在实施本申请方案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。也可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。本领域普通技术人员在不付出创造性劳动的情况下,即可以理解并实施。

[0095] 以上所述仅是本申请的具体实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本申请原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本申请的保护范围。

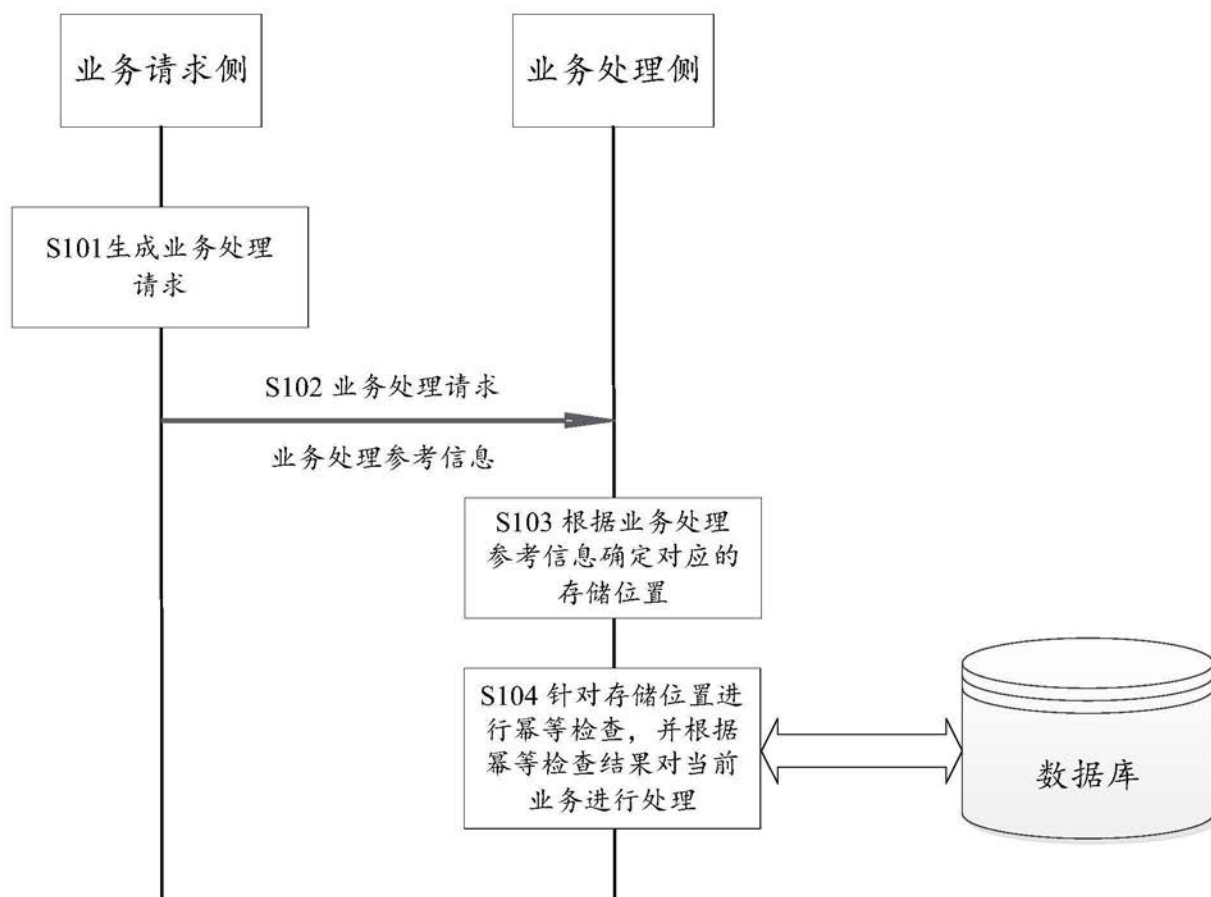


图1

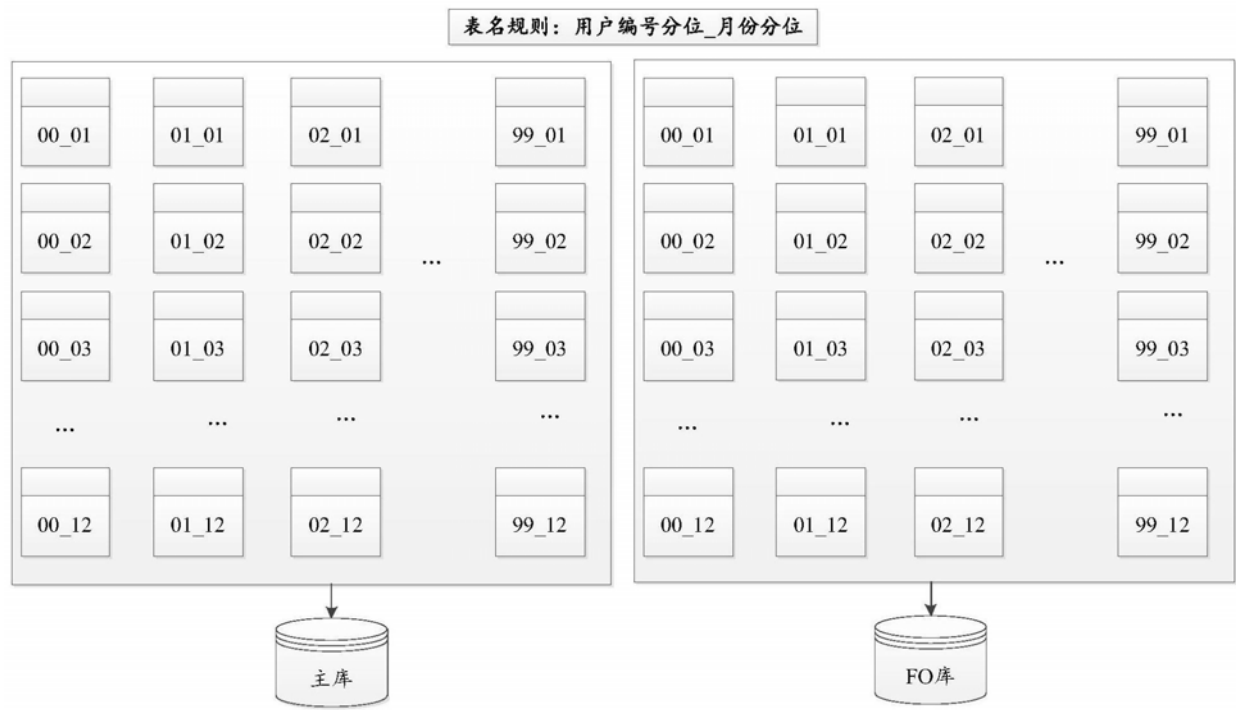


图2

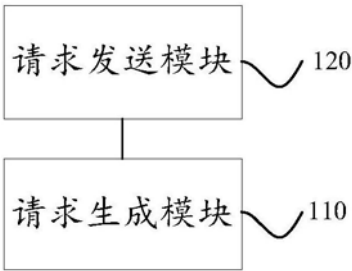


图3

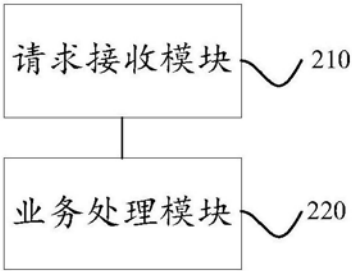


图4

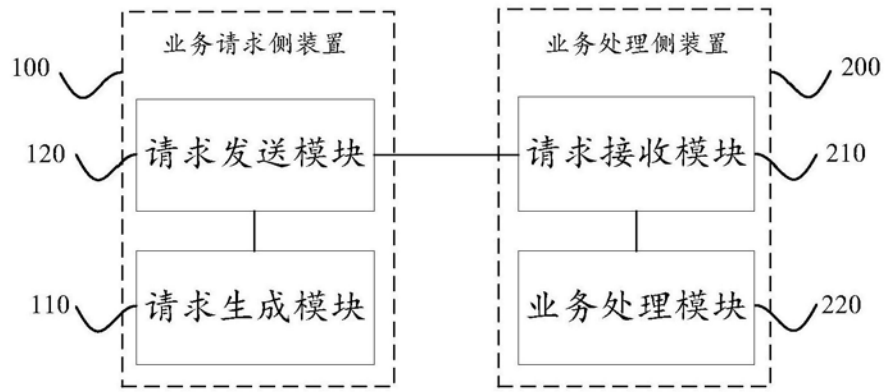


图5