



(12) 发明专利

(10) 授权公告号 CN 112131237 B

(45) 授权公告日 2024. 06. 14

(21) 申请号 202011044851.8

(56) 对比文件

(22) 申请日 2020.09.28

CN 111563102 A, 2020.08.21

(65) 同一申请的已公布的文献号

审查员 徐晓

申请公布号 CN 112131237 A

(43) 申请公布日 2020.12.25

(73) 专利权人 京东科技控股股份有限公司

地址 100176 北京市大兴区北京经济技术

开发区科创十一街18号C座2层221室

(72) 发明人 孙亮

(74) 专利代理机构 北京华夏泰和知识产权代理

有限公司 11662

专利代理师 曾军 刘晓燕

(51) Int. Cl.

G06F 16/23 (2019.01)

G06F 16/27 (2019.01)

权利要求书2页 说明书10页 附图3页

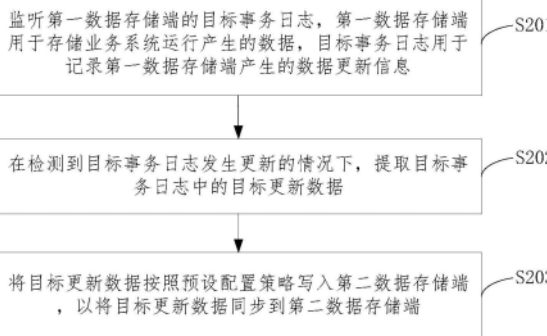
(54) 发明名称

数据同步方法、装置、设备及计算机可读介

质

(57) 摘要

本申请涉及一种数据同步方法、装置、设备及计算机可读介质。该方法包括：监听第一数据存储端的目标事务日志，第一数据存储端用于存储业务系统运行产生的数据，目标事务日志用于记录第一数据存储端产生的数据更新信息；在检测到目标事务日志发生更新的情况下，提取目标事务日志中的目标更新数据；将目标更新数据按照预设配置策略写入第二数据存储端，以将目标更新数据同步到第二数据存储端。本申请可以通过源数据存储端的二进制日志文件来确定哪些数据发生了更新及更新方式，从而目标数据存储端可以不需要业务功能模块的指示直接进行数据同步，完全将数据同步与业务功能解耦合，便于两个系统的维护及迭代且不会互相影响稳定性。



1. 一种数据同步方法,其特征在于,包括:

监听第一数据存储端的目标事务日志,其中,所述第一数据存储端用于存储业务系统运行产生的数据,所述目标事务日志用于记录所述第一数据存储端产生的数据更新信息,所述目标事务日志为二进制日志;

在检测到所述目标事务日志发生更新的情况下,提取所述目标事务日志中的目标更新数据;

将所述目标更新数据按照预设配置策略写入第二数据存储端,以将所述目标更新数据同步到所述第二数据存储端;

在将所述目标更新数据按照预设配置策略写入第二数据存储端时发生异常的情况下,所述方法还包括:利用第一函数捕获第二函数的异常,其中,所述第一函数为所述第二函数的外层函数,所述第二函数用于将所述目标更新数据按照所述预设配置策略写入所述第二数据存储端;继续利用所述第一函数将所述目标更新数据按照所述预设配置策略写入所述第二数据存储端,直至消除异常。

2. 根据权利要求1所述的方法,其特征在于,将所述目标更新数据按照预设配置策略写入第二数据存储端之前,所述方法包括按照如下方式设置所述预设配置策略:

从多个待选数据存储端中确定目标数据存储端作为所述第二数据存储端;

配置与所述第二数据存储端匹配的目标服务集群,并建立与所述第二数据存储端匹配的目标索引。

3. 根据权利要求2所述的方法,其特征在于,建立所述目标索引之后,所述方法还包括:

建立目标源数据与目标存储数据的映射关系,其中,所述目标源数据为所述第一数据存储端中的所述目标更新数据,所述目标存储数据为同步到所述第二数据存储端的数据;

利用内存管理系统将所述映射关系加载到内存中。

4. 根据权利要求3所述的方法,其特征在于,建立目标源数据与目标存储数据的映射关系包括:

确定所述目标存储数据的存储格式、存储路径及版本控制字段;

采用目标表达式语言,将所述目标源数据按照所述存储格式、所述存储路径及所述版本控制字段进行编码。

5. 根据权利要求3所述的方法,其特征在于,将所述目标更新数据按照预设配置策略写入第二数据存储端包括:

按照所述映射关系将所述目标更新数据转换为所述目标存储数据;

将所述目标存储数据存储至所述第二数据存储端。

6. 根据权利要求4所述的方法,其特征在于,将所述目标更新数据按照预设配置策略写入第二数据存储端还包括:

在进行存量数据同步的情况下,确定所述存量数据的当前版本字段,其中,所述目标更新数据包括所述存量数据;

在所述第二数据存储端未查找到大于所述当前版本字段的所述版本控制字段的情况下,将所述存量数据按照所述当前版本字段存储至所述第二数据存储端。

7. 一种数据同步装置,其特征在于,包括:

日志监听模块,用于监听第一数据存储端的目标事务日志,其中,所述第一数据存储端

用于存储业务系统运行产生的数据,所述目标事务日志用于记录所述第一数据存储端产生的数据更新信息,所述目标事务日志为二进制日志;

数据提取模块,用于在检测到所述目标事务日志发生更新的情况下,提取所述目标事务日志中的目标更新数据;

数据同步模块,用于将所述目标更新数据按照预设配置策略写入第二数据存储端,以将所述目标更新数据同步到所述第二数据存储端;

所述数据同步模块,还用于:在将所述目标更新数据按照预设配置策略写入第二数据存储端时发生异常的情况下,利用第一函数捕获第二函数的异常,其中,所述第一函数为所述第二函数的外层函数,所述第二函数用于将所述目标更新数据按照所述预设配置策略写入所述第二数据存储端;继续利用所述第一函数将所述目标更新数据按照所述预设配置策略写入所述第二数据存储端,直至消除异常。

8.一种电子设备,包括存储器、处理器、通信接口及通信总线,所述存储器中存储有可在所述处理器上运行的计算机程序,所述存储器、所述处理器通过所述通信总线和所述通信接口进行通信,其特征在于,所述处理器执行所述计算机程序时实现上述权利要求1至6任一项所述的方法的步骤。

9.一种具有处理器可执行的非易失的程序代码的计算机可读介质,其特征在于,所述程序代码使所述处理器执行所述权利要求1至6任一所述方法。

数据同步方法、装置、设备及计算机可读介质

技术领域

[0001] 本申请涉及数据处理技术领域,尤其涉及一种数据同步方法、装置、设备及计算机可读介质。

背景技术

[0002] 数据同步是指将数据从一个存储媒介同步到另一个存储媒介,两个存储媒介可能相同或不同,比如从MySQL到Elasticsearch、从MySQL到HBase、从MySQL到Elasticsearch和HBase等,数据同步可以是内部业务逻辑触发或是外部触发。业务逻辑触发:例如当数据bizType=1的时候将数据从一个MySQL数据库同步到另一个MySQL数据库,当数据bizType=2的时候将数据从MySQL同步到HBase。外部触发:例如每天凌晨3点定时任务将MySQL某个表的全量数据同步到Elasticsearch的某个index中。

[0003] 目前,相关技术中,对要进行同步的数据,一般是采用双写的方式,例如业务模块向MySQL写入数据,因为需要将数据同步到Elasticsearch,业务模块在写MySQL成功之后调用api写入Elasticsearch。这种方式往往使得业务功能与数据同步功能具有较高的耦合性。

[0004] 针对上述的问题,目前尚未提出有效的解决方案。

发明内容

[0005] 本申请提供了一种数据同步方法、装置、设备及计算机可读介质,以解决数据同步功能和业务功能耦合度高的技术问题。

[0006] 根据本申请实施例的一个方面,本申请提供了一种数据同步方法,包括:监听第一数据存储端的目标事务日志,第一数据存储端用于存储业务系统运行产生的数据,目标事务日志用于记录第一数据存储端产生的数据更新信息;在检测到目标事务日志发生更新的情况下,提取目标事务日志中的目标更新数据;将目标更新数据按照预设配置策略写入第二数据存储端,以将目标更新数据同步到第二数据存储端。

[0007] 可选地,将目标更新数据按照预设配置策略写入第二数据存储端之前,该方法还包括按照如下方式设置预设配置策略:从多个待选数据存储端中确定目标数据存储端作为第二数据存储端;配置与第二数据存储端匹配的目标服务集群,并建立与第二数据存储端匹配的目标索引。

[0008] 可选地,建立目标索引之后,该方法还包括:建立目标源数据与目标存储数据的映射关系,目标源数据为第一数据存储端中的目标更新数据,目标存储数据为同步到第二数据存储端的数据;利用内存管理系统将映射关系加载到内存中。

[0009] 可选地,建立目标源数据与目标存储数据的映射关系包括:确定目标存储数据的存储格式、存储路径及版本控制字段;采用目标表达式语言,将目标源数据按照存储格式、存储路径及版本控制字段进行编码。

[0010] 可选地,将目标更新数据按照预设配置策略写入第二数据存储端包括:按照映射

关系将目标更新数据转换为目标存储数据;将目标存储数据存储至第二数据存储端。

[0011] 可选地,将目标更新数据按照预设配置策略写入第二数据存储端还包括:在进行存量数据同步的情况下,确定存量数据的当前版本字段,目标更新数据包括存量数据;在第二数据存储端未查找到大于当前版本字段的版本控制字段的情况下,将存量数据按照当前版本字段存储至第二数据存储端。

[0012] 可选地,在将目标更新数据按照预设配置策略写入第二数据存储端时发生异常的情况下,该方法还包括:利用第一函数捕获第二函数的异常,第一函数为第二函数的外层函数,第二函数用于将目标更新数据按照预设配置策略写入第二数据存储端;继续利用第一函数将目标更新数据按照预设配置策略写入第二数据存储端,直至消除异常。

[0013] 根据本申请实施例的另一方面,本申请提供了一种数据同步装置,包括:日志监听模块,用于监听第一数据存储端的目标事务日志,第一数据存储端用于存储业务系统运行产生的数据,目标事务日志用于记录第一数据存储端产生的数据更新信息;数据提取模块,用于在检测到目标事务日志发生更新的情况下,提取目标事务日志中的目标更新数据;数据同步模块,用于将目标更新数据按照预设配置策略写入第二数据存储端,以将目标更新数据同步到第二数据存储端。

[0014] 根据本申请实施例的另一方面,本申请提供了一种电子设备,包括存储器、处理器、通信接口及通信总线,存储器中存储有可在处理器上运行的计算机程序,存储器、处理器通过通信总线和通信接口进行通信,处理器执行计算机程序时实现上述方法的步骤。

[0015] 根据本申请实施例的另一方面,本申请还提供了一种具有处理器可执行的非易失的程序代码的计算机可读介质,程序代码使处理器执行上述的方法。

[0016] 本申请实施例提供的上述技术方案与相关技术相比具有如下优点:

[0017] 本申请技术方案为监听第一数据存储端的目标事务日志,第一数据存储端用于存储业务系统运行产生的数据,目标事务日志用于记录第一数据存储端产生的数据更新信息;在检测到目标事务日志发生更新的情况下,提取目标事务日志中的目标更新数据;将目标更新数据按照预设配置策略写入第二数据存储端,以将目标更新数据同步到第二数据存储端。本申请可以通过源数据存储端的二进制日志文件来确定哪些数据发生了更新及更新方式,从而目标数据存储端可以不需要业务功能模块的指示直接进行数据同步,完全将数据同步与业务功能解耦合,便于两个系统的维护及迭代且不会互相影响稳定性,并且,还能通过重试机制保证数据最终一致性。

附图说明

[0018] 此处的附图被并入说明书中并构成本说明书的一部分,示出了符合本申请的实施例,并与说明书一起用于解释本申请的原理。

[0019] 为了更清楚地说明本申请实施例或相关技术中的技术方案,下面将对实施例或相关技术描述中所需要使用的附图作简单地介绍,显而易见地,对于本领域普通技术人员而言,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0020] 图1为根据本申请实施例提供的一种可选的数据同步方法硬件环境示意图;

[0021] 图2为根据本申请实施例提供的一种可选的数据同步方法流程图;

[0022] 图3为根据本申请实施例提供的一种可选的配置方法流程图;

- [0023] 图4为根据本申请实施例提供的一种可选的配置方法流程图;
- [0024] 图5为根据本申请实施例提供的一种可选的配置方法流程图;
- [0025] 图6为根据本申请实施例提供的一种可选的数据同步方法流程图;
- [0026] 图7为根据本申请实施例提供的一种可选的数据同步装置框图;
- [0027] 图8为本申请实施例提供的一种可选的电子设备结构示意图。

具体实施方式

[0028] 为使本申请实施例的目的、技术方案和优点更加清楚,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本申请的一部分实施例,而不是全部的实施例。基于本申请中的实施例,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,都属于本申请保护的范围。

[0029] 在后续的描述中,使用用于表示元件的诸如“模块”、“部件”或“单元”的后缀仅为了有利于本申请的说明,其本身并没有特定的意义。因此,“模块”与“部件”可以混合地使用。

[0030] 首先,在对本申请实施例进行描述的过程中出现的部分名词或者术语适用于如下解释:

[0031] binlog:用于MySQL主从同步,记录了对MySQL数据库执行更改的所有操作(不包括SELECT和SHOW这样的操作),若操作本身并没有导致数据库发生变化,那么该操作也会写入二进制日志文件中。

[0032] Elasticsearch:Elasticsearch是一个基于Lucene的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎,基于RESTful web接口,用Java语言开发的,并作为Apache许可条款下的开放源码发布,是一种流行的企业级搜索引擎。

[0033] HBase:一个分布式的、面向列的开源数据库,该技术来源于Fay Chang所撰写的Google论文“Bigtable:一个结构化数据的分布式存储系统”。就像Bigtable利用了Google文件系统(File System)所提供的分布式数据存储一样,HBase在Hadoop之上提供了类似于Bigtable的能力,是Apache的Hadoop项目的子项目。

[0034] spel:全称是Spring Expression Language,即spring提供的表达式语言,可以在运行期进行查询和操作数据,提供了丰富的计算及操作方式,便于在程序中执行配置性操作。

[0035] 相关技术中,对要进行同步的数据,一般是采用双写的方式,例如业务模块向MySQL写入数据,因为需要将数据同步到Elasticsearch,业务模块在写MySQL成功之后调用api写入Elasticsearch。这种方式往往使得业务功能与数据同步功能具有较高的耦合性,业务功能与数据同步功能耦合在一起,使业务系统的逻辑更复杂化,提升了后续开发及维度的难度,并且会导致业务系统功能与数据同步功能互相影响导致可用性下降。不仅如此,相关技术中还存在以下几个问题:

[0036] 由于相关技术方案的逻辑是写MySQL成功后再写入Elasticsearch,若写入Elasticsearch时失败,由于MySQL已经成功写入,此时MySQL与Elasticsearch数据不一致,若要在此方案中解决事务问题会引入更复杂的方案及相应问题。

[0037] 相关技术方案都是通过硬编码实现,所以表之间的关系及字段映射都是固定的,

例如原始需求是MySQL的table1表同步到Elasticsearch的index1索引、则MySQL与Elasticsearch的字段是一一对应的,不易扩展。

[0038] 相关技术方案中未考虑存量数据的同步问题,即只同步上线之后产生的数据,并未考虑上线前产生的存量数据,此时若有对存量数据的同步诉求则无法实现。

[0039] 为了解决背景技术中提及的问题,根据本申请实施例的一方面,提供了一种数据同步方法的实施例。

[0040] 可选地,在本申请实施例中,上述数据同步方法可以应用于如图1所示的由终端101和服务器102所构成的硬件环境中。如图1所示,服务器102通过网络与终端101进行连接,可用于为终端或终端上安装的客户端提供服务,可在服务器上或独立于服务器设置数据库103,用于为服务器102提供数据存储服务,上述网络包括但不限于:广域网、城域网或局域网,终端101包括但不限于PC、手机、平板电脑等。

[0041] 本申请实施例中的一种数据同步方法可以由服务器102来执行,还可以是由服务器102和终端101共同执行,如图2所示,该方法可以包括以下步骤:

[0042] 步骤S201,监听第一数据存储端的目标事务日志,第一数据存储端用于存储业务系统运行产生的数据,目标事务日志用于记录第一数据存储端产生的数据更新信息。

[0043] 本申请实施例中,第一数据存储端可以是源数据存储端,即需要同步数据的一端,第二数据存储端可以是目标数据存储端,即将要同步数据的一端。源数据存储端中需要同步的数据为目标源数据,同步到了目标数据存储端的数据为目标存储数据。目标源数据和目标存储数据可以是完全一致的数据,也可以是根据实际情况或需要按照配置策略进行数据同步。

[0044] 本申请实施例中,对于数据同步前的数据接入,可以通过目标事务日志实现。目标事务日志可以是一种数据库的日志类型,以MySQL为例,MySQL中一般有以下几种类型的日志:

[0045] 错误日志,用于记录在启动、运行或停止MySQL时遇到的问题;

[0046] 通用查询日志,用于记录建立的客户端连接和执行的语句;

[0047] 二进制日志(binlog),用于记录更改数据的语句;

[0048] 中继日志,用于复制主数据库接收的数据更改;

[0049] 慢查询日志,用于记录所有执行时间超过查询时间阈值的所有查询或不使用索引的查询。

[0050] 作为优选,在本申请实施例中可以以二进制日志(binlog)作为目标事务日志,二进制日志以事件形式记录数据库中数据发生的更改,还包含语句所执行的消耗的时间。二进制日志的格式类型可以是STATEMENT、ROW及MIXED。

[0051] STATEMENT,即基于SQL语句的复制,记录的是修改SQL语句,优点是日志文件小,节约输入/输出(I/O)资源,性能较高,缺点是由于记录的只是执行语句,为了这些语句能在从数据库上正确运行,因此还必须记录每条语句在执行的时候的一些相关信息,以保证所有语句能在从数据库得到和在主数据库执行的时候相同的结果。

[0052] ROW,即基于行的复制,该类型不记录SQL语句上下文相关信息,仅保存哪条记录被修改。优点是基于ROW的日志内容会非常清楚的记录下每一行数据修改的细节。而且不会出现某些特定情况下的存储过程,或function,以及trigger的调用和触发无法被正确复制的

问题,缺点是所有的执行的语句当记录到日志中的时候,都将以每行记录的修改来记录,这样可能会产生大量的日志内容。

[0053] MIXED,即STATEMENT和ROW的混合模式复制。在MIXED模式下,一般的语句修改使用STATEMENT格式保存二进制日志,如一些函数,STATEMENT无法完成主从复制的操作,则采用ROW格式保存二进制日志,MySQL会根据执行的每一条具体的SQL语句来区分对待记录的日志形式,也就是在STATEMENT和ROW之间选择一种。

[0054] 本申请实施例中,可以根据源数据存储端中采用的二进制日志文件格式,选择相应的格式进行数据同步处理。

[0055] 步骤S202,在检测到目标事务日志发生更新的情况下,提取目标事务日志中的目标更新数据。

[0056] 本申请实施例中,二进制日志,即目标事务日志可以记录下源数据存储端的数据发生的更改,因此可以在监听源数据存储端(即第一数据存储端)的二进制日志文件,在源数据存储端的二进制日志文件更新的情况下,说明源数据存储端的数据发生了更改,为了将源数据存储端与目标数据存储端的数据同步,可以只通过源数据存储端的二进制日志文件提取数据更新记录,并提取更新的数据,从而可以进行后续的将更新数据从源数据存储端同步到目标数据存储端的步骤。

[0057] 步骤S203,将目标更新数据按照预设配置策略写入第二数据存储端,以将目标更新数据同步到第二数据存储端。

[0058] 本申请实施例中,可以将提取到的更新数据按照目标数据存储端(即第二数据存储端)的配置策略同步到目标数据存储端。

[0059] 本申请实施例中,以数据同步的源和目标都是数据库为例进行说明,可以通过二进制日志获取源数据库中数据发生的改变,再将这些数据发生的改变同步到目标数据库中,从而不再需要业务功能模块发出数据同步指令并进行相应操作,实现将业务功能模块和数据同步模块解耦合,使得业务功能模块不再需要参与数据同步工作。

[0060] 本申请实施例中,对于数据同步前的数据接入,还可以采用消息队列接入和JavaServer Faces框架接入的方式。消息队列接入的方式优点是天然异步,大流量场景可以用消息队列做消费缓冲,不会有因压力大导致某个数据同步系统异常的风险。JavaServer Faces框架接入的方式接入程度比较低,但如果要做异步方式需自己使用线程池实现,缺点是无法缓冲,可能因为请求量大导致JavaServer Faces线程池耗尽或是消费端异常。

[0061] 采用本申请技术方案,可以通过源数据存储端的二进制日志文件来确定哪些数据发生了更新及更新方式,从而目标数据存储端可以不需要业务功能模块的指示直接进行数据同步,完全将数据同步与业务功能解耦合,便于两个系统的维护及迭代且不会互相影响稳定性。

[0062] 可选地,将目标更新数据按照预设配置策略写入第二数据存储端之前,如图3所示,该方法还可以包括按照如下方式设置预设配置策略:

[0063] 步骤S301,从多个待选数据存储端中确定目标数据存储端作为第二数据存储端。

[0064] 本申请实施中,上述待选数据存储端可以是MySQL数据库,可以是HBase数据库,还可以是Elasticsearch全文搜索引擎等。可以根据实际需要选择相应的目标数据存储端。

[0065] 步骤S302,配置与第二数据存储端匹配的目标服务集群,并建立与第二数据存储端匹配的目标索引。

[0066] 本申请实施中,以Elasticsearch为目标数据存储端为例,在进行数据同步之前,可以申请及配置Elasticsearch集群、索引。配置集群可以提高系统性能,避免数据同步过程中一个服务器宕机导致整个系统奔溃的问题,还可以降低成本、提高可扩展性,并增强可靠性。配置索引是为了不通过存取整张表的数据,而是搜索已经排序的索引,然后通过索引定位到表中相应的数据,从而快速搜索到需要获取的条目。

[0067] 可选地,建立匹配的服务集群和索引之后,还需要建立源数据存储端到目标数据存储端之间的数据映射关系,如图4所示,具体步骤可以是:

[0068] 步骤S401,建立目标源数据与目标存储数据的映射关系,目标源数据为第一数据存储端中的目标更新数据,目标存储数据为同步到第二数据存储端的数据。

[0069] 本申请实施例中,源数据存储端存储的是目标源数据,即需要同步的数据,目标数据存储端存储的是目标存储数据,即同步过来的数据,目标源数据和目标存储数据可以是完全一致的,也可以根据实际情况或需要进行改动,例如,可以根据需要配置计算策略,对目标源数据赋予权重,求加权和,还可以再数据同步的过程中去除冗余数据,只保留重要数据。

[0070] 步骤S402,利用内存管理系统将映射关系加载到内存中。

[0071] 本申请实施例中,为了将目标源数据按照映射关系转换为目标存储数据,可以通过内存管理器将映射关系及其他配置信息加载到内存中,从而实时对目标源数据按照映射关系和配置信息进行数据同步。

[0072] 本申请实施例中,可以采用对象池模式,根据实际情况或需求,在进行配置时,复用池中对象,没有分配内存和创建堆中对象的开销,没有释放内存和销毁堆中对象的开销,进而减少垃圾收集器的负担,避免内存抖动,不必重复初始化对象状态,能够有效提高性能。其中,具体的配置信息可以是抽数配置、写数配置、异常处理配置等。

[0073] 本申请实施例中,还可以根据需要同步的数据表及数据表中需要同步的信息适应性的配置相关模块,从而新增数据同步需求时只需要简单配置即可,不需要开发。采用本申请技术方案,能够极大的便于功能扩展,降低后续的开发及维护工作量。

[0074] 可选地,如图5所示,建立目标源数据与目标存储数据的映射关系可以包括:

[0075] 步骤S501,确定目标存储数据的存储格式、存储路径及版本控制字段。

[0076] 步骤S502,采用目标表达式语言,将目标源数据按照存储格式、存储路径及版本控制字段进行编码。

[0077] 本申请实施例中,可以根据不同的目标数据存储端,确定不同的数据存储格式、存储路径和版本信息等。可以采用Spring Expression Language进行字段映射解析及特殊值计算,例如,“applydate”:“#{tf(map[applydate])}”表示源数据存储端的applydate字段通过自定义的tf方式转换为目标数据存储端的applydate字段,相似的,根据需要配置计算策略,对目标源数据赋予权重,求加权和的计算方式的配置可以是:“totalcount”:“#{orderCount}+#{amountCount}”。

[0078] 可选地,如图6所示,在对离线数据(即存量数据)进行数据同步时,还可以包括以下步骤:

[0079] 步骤S601,在进行存量数据同步的情况下,确定存量数据的当前版本字段,目标更新数据包括存量数据。

[0080] 步骤S602,在第二数据存储端未查找到大于当前版本字段的版本控制字段的情况下,将存量数据按照当前版本字段存储至第二数据存储端。

[0081] 本申请实施例中,数据从时效性方面即分为离线数据和实时数据,离线数据即上线前的存量数据,实时数据即上线后收到的流式数据。离线数据的特点是基量较大,所以在做离线数据同步时首要保证的就是性能和稳定性,性能即快速完成全部基量数据的同步工作,稳定性即保证任务可以正确终止,不能因为内存溢出等问题导致任务半途终止。

[0082] 离线数据和实时数据并没有严格的界限,例如10号上线,10号之前产生的数据称为离线数据,但因为数据是可以变更的,所以也可能在10号或之后收到变更后的实时数据,即在做离线数据同步时也要考虑离线数据与实时数据之间的顺序问题,本方案通过版本字段来表示数据的更新顺序,版本字段越大,则表示数据更新的时间越晚,例如可以配置offVersionKey进行限制,当离线数据的ofVersionKey对应的值比已经存储的值小的时候就忽略当前离线数据。采用本申请技术方案,精度可以精确到秒。

[0083] 可选地,在将目标更新数据按照预设配置策略写入第二数据存储端时发生异常的情况下,该方法还包括:利用第一函数捕获第二函数的异常,第一函数为第二函数的外层函数,第二函数用于将目标更新数据按照预设配置策略写入第二数据存储端;继续利用第一函数将目标更新数据按照预设配置策略写入第二数据存储端,直至消除异常。

[0084] 本申请实施例中,在数据同步过程中发生异常如网络连接不稳定等情况时,可以将异常抛出到外层函数,由外层函数继续重试直至数据同步完成。内层函数抛出异常,由外层函数抓取异常并进行处理,可以极大简化内层核心功能的代码量,减少系统奔溃的概率,从而可以通过重试机制,实现数据最终一致性。

[0085] 本申请技术方案为监听第一数据存储端的目标事务日志,第一数据存储端用于存储业务系统运行产生的数据,目标事务日志用于记录第一数据存储端产生的数据更新信息;在检测到目标事务日志发生更新的情况下,提取目标事务日志中的目标更新数据;将目标更新数据按照预设配置策略写入第二数据存储端,以将目标更新数据同步到第二数据存储端。本申请可以通过源数据存储端的二进制日志文件来确定哪些数据发生了更新及更新方式,从而目标数据存储端可以不需要业务功能模块的指示直接进行数据同步,完全将数据同步与业务功能解耦合,便于两个系统的维护及迭代且不会互相影响稳定性,并且,还能通过重试机制保证数据最终一致性。

[0086] 根据本申请实施例的又一方面,如图7所示,提供了一种数据同步装置,包括:日志监听模块701,用于监听第一数据存储端的目标事务日志,第一数据存储端用于存储业务系统运行产生的数据,目标事务日志用于记录第一数据存储端产生的数据更新信息;数据提取模块702,用于在检测到目标事务日志发生更新的情况下,提取目标事务日志中的目标更新数据;数据同步模块703,用于将目标更新数据按照预设配置策略写入第二数据存储端,以将目标更新数据同步到第二数据存储端。

[0087] 需要说明的是,该实施例中的日志监听模块701可以用于执行本申请实施例中的步骤S201,该实施例中的数据提取模块702可以用于执行本申请实施例中的步骤S202,该实施例中的数据同步模块703可以用于执行本申请实施例中的步骤S203。

[0088] 此处需要说明的是,上述模块与对应的步骤所实现的示例和应用场景相同,但不限于上述实施例所公开的内容。需要说明的是,上述模块作为装置的一部分可以运行在如图1所示的硬件环境中,可以通过软件实现,也可以通过硬件实现。

[0089] 可选地,该数据同步装置,还包括配置模块,用于:从多个待选数据存储端中确定目标数据存储端作为第二数据存储端;配置与第二数据存储端匹配的目标服务集群,并建立与第二数据存储端匹配的目标索引。

[0090] 可选地,该数据同步装置,还包括映射模块,用于:建立目标源数据与目标存储数据的映射关系,目标源数据为第一数据存储端中的目标更新数据,目标存储数据为同步到第二数据存储端的数据;利用内存管理系统将映射关系加载到内存中。

[0091] 可选地,该映射模块,还用于:确定目标存储数据的存储格式、存储路径及版本控制字段;采用目标表达式语言,将目标源数据按照存储格式、存储路径及版本控制字段进行编码。

[0092] 可选地,该数据同步模块,具体用于:按照映射关系将目标更新数据转换为目标存储数据;将目标存储数据存储至第二数据存储端。

[0093] 可选地,该数据同步模块,还用于:在进行存量数据同步的情况下,确定存量数据的当前版本字段,目标更新数据包括存量数据;在第二数据存储端未查找到大于当前版本字段的版本控制字段的情况下,将存量数据按照当前版本字段存储至第二数据存储端。

[0094] 可选地,该数据同步装置,还包括异常处理模块,用于:利用第一函数捕获第二函数的异常,第一函数为第二函数的外层函数,第二函数用于将目标更新数据按照预设配置策略写入第二数据存储端;继续利用第一函数将目标更新数据按照预设配置策略写入第二数据存储端,直至消除异常。

[0095] 根据本申请实施例的另一方面,本申请提供了一种电子设备,如图8所示,包括存储器801、处理器802、通信接口803及通信总线804,存储器801中存储有可在处理器802上运行的计算机程序,存储器801、处理器802通过通信接口803和通信总线804进行通信,处理器802执行计算机程序时实现上述方法的步骤。

[0096] 上述电子设备中的存储器、处理器通过通信总线和通信接口进行通信。所述通信总线可以是外设部件互连标准(Peripheral Component Interconnect,简称PCI)总线或扩展工业标准结构(Extended Industry Standard Architecture,简称EISA)总线等。该通信总线可以分为地址总线、数据总线、控制总线等。

[0097] 存储器可以包括随机存取存储器(Random Access Memory,简称RAM),也可以包括非易失性存储器(non-volatile memory),例如至少一个磁盘存储器。可选的,存储器还可以是至少一个位于远离前述处理器的存储装置。

[0098] 上述的处理器可以是通用处理器,包括中央处理器(Central Processing Unit,简称CPU)、网络处理器(Network Processor,简称NP)等;还可以是数字信号处理器(Digital Signal Processing,简称DSP)、专用集成电路(Application Specific Integrated Circuit,简称ASIC)、现场可编程门阵列(Field-Programmable Gate Array,简称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。

[0099] 根据本申请实施例的又一方面还提供了一种具有处理器可执行的非易失的程序代码的计算机可读介质。

[0100] 可选地,在本申请实施例中,计算机可读介质被设置为存储用于所述处理器执行以下步骤的程序代码:

[0101] 监听第一数据存储端的目标事务日志,第一数据存储端用于存储业务系统运行产生的数据,目标事务日志用于记录第一数据存储端产生的数据更新信息;

[0102] 在检测到目标事务日志发生更新的情况下,提取目标事务日志中的目标更新数据;

[0103] 将目标更新数据按照预设配置策略写入第二数据存储端,以将目标更新数据同步到第二数据存储端。

[0104] 可选地,本实施例中的具体示例可以参考上述实施例中所描述的示例,本实施例在此不再赘述。

[0105] 本申请实施例在具体实现时,可以参阅上述各个实施例,具有相应的技术效果。

[0106] 可以理解的是,本文描述的这些实施例可以用硬件、软件、固件、中间件、微码或其组合来实现。对于硬件实现,处理单元可以实现在一个或多个专用集成电路(Application Specific Integrated Circuits,ASIC)、数字信号处理器(Digital Signal Processing,DSP)、数字信号处理设备(DSP Device,DSPD)、可编程逻辑设备(Programmable Logic Device,PLD)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)、通用处理器、控制器、微控制器、微处理器、用于执行本申请所述功能的其它电子单元或其组合中。

[0107] 对于软件实现,可通过执行本文所述功能的单元来实现本文所述的技术。软件代码可存储在存储器中并通过处理器执行。存储器可以在处理器中或在处理器外部实现。

[0108] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。

[0109] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统、装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0110] 在本申请所提供的实施例中,应该理解到,所揭露的装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述模块的划分,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,例如多个模块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。

[0111] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0112] 另外,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单元中,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单元中。

[0113] 所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以

存储在一个计算机可读取存储介质中。基于这样的理解,本申请实施例的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,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所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。需要说明的是,在本文中,诸如“第一”和“第二”等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0114] 以上所述仅是本申请的具体实施方式,使本领域技术人员能够理解或实现本申请。对这些实施例的多种修改对本领域的技术人员来说将是显而易见的,本文中所定义的一般原理可以在不脱离本申请的精神或范围的情况下,在其它实施例中实现。因此,本申请将不会被限制于本文所示的这些实施例,而是要符合与本文所申请的原理和新颖特点相一致的最宽的范围。

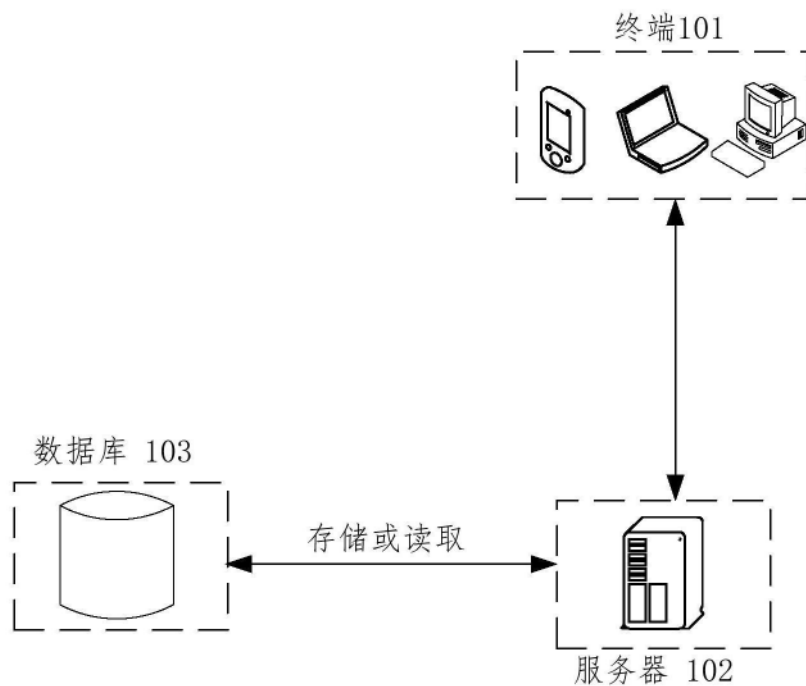


图1

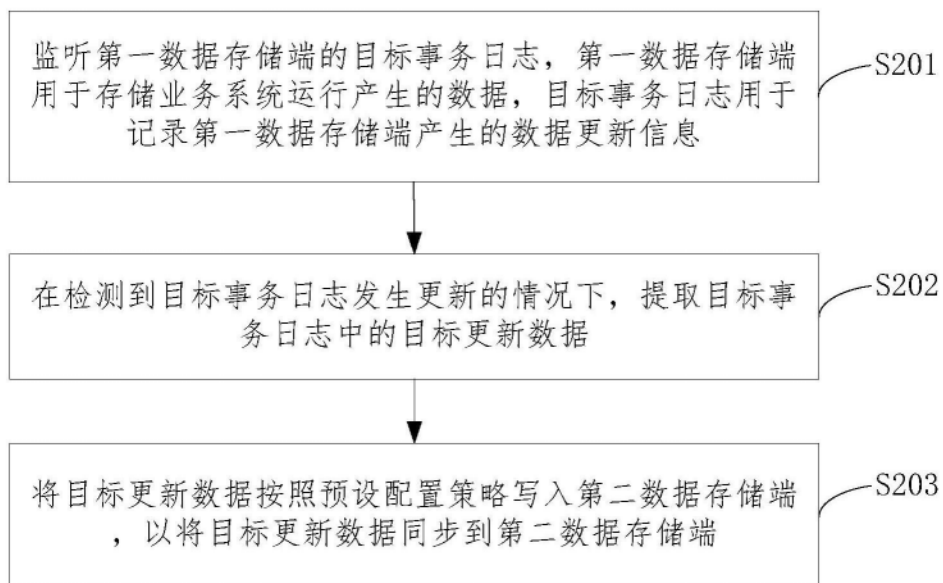


图2

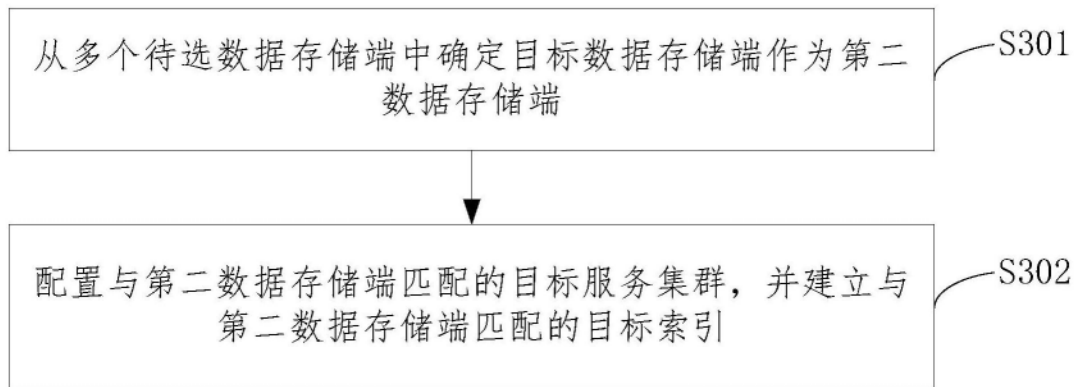


图3

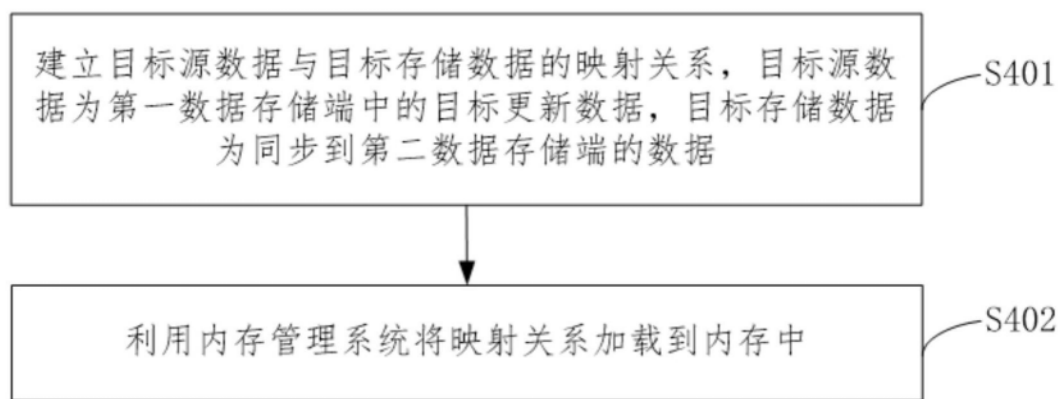


图4

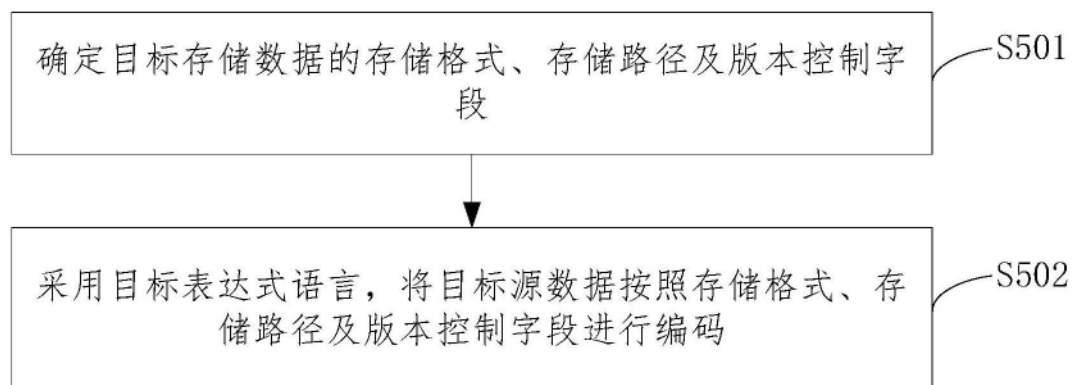


图5

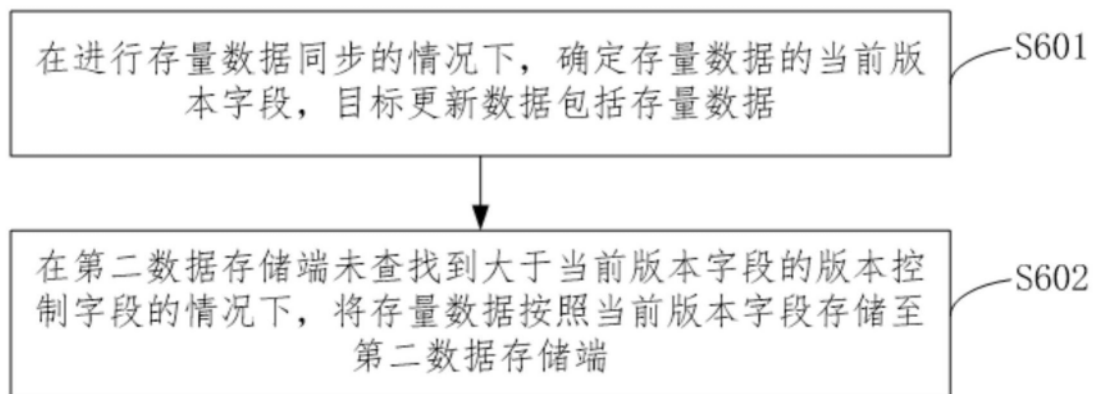


图6

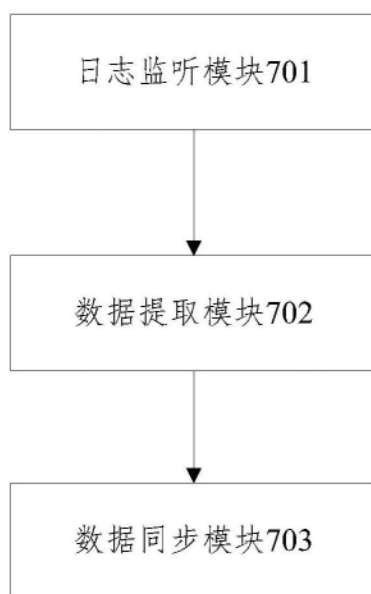


图7

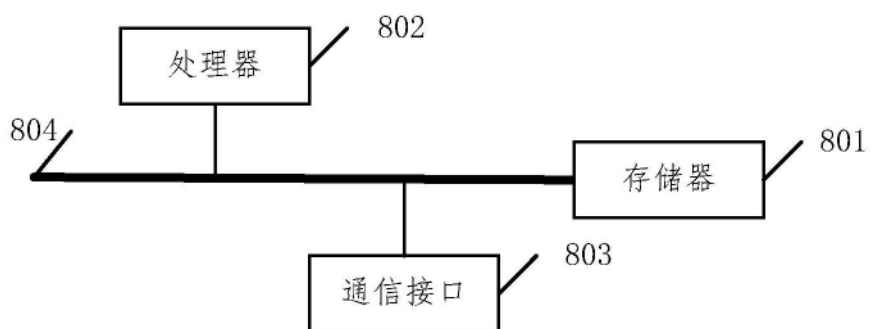


图8