



(12) 发明专利

(10) 授权公告号 CN 117763052 B

(45) 授权公告日 2024. 05. 10

(21) 申请号 202410196438.5

G06Q 50/60 (2024.01)

(22) 申请日 2024.02.22

(56) 对比文件

(65) 同一申请的已公布的文献号

申请公布号 CN 117763052 A

CN 102999584 A, 2013.03.27

CN 109783012 A, 2019.05.21

CN 104793988 A, 2015.07.22

(43) 申请公布日 2024.03.26

CN 107273542 A, 2017.10.20

CN 113886403 A, 2022.01.04

(73) 专利权人 浩鲸云计算科技股份有限公司

地址 210000 江苏省南京市雨花台区安德

门大街50号B座6楼

CN 114079660 A, 2022.02.22

CN 116610752 A, 2023.08.18

(72) 发明人 汲洋弘康 林李杰 李静

US 2004026496 A1, 2004.02.12

US 2020364241 A1, 2020.11.19

(74) 专利代理机构 南京中盟科创知识产权代理

事务所(特殊普通合伙)

32279

专利代理师 孙丽君

US 9697226 B1, 2017.07.04

杨海涛等. 大规模云同步归集数据系统的异步并行优化. 计算机工程与应用. 2017, 第53卷(第02期), 88-97.

(51) Int. Cl.

G06F 16/27 (2019.01)

G06Q 30/0283 (2023.01)

审查员 袁勤

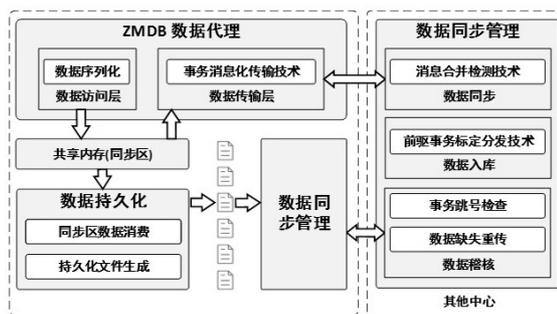
权利要求书2页 说明书14页 附图8页

(54) 发明名称

面向计费多中心内存数据库的数据同步方法及系统

(57) 摘要

本发明提出面向计费多中心内存数据库的数据同步方法及系统,包括:ZMDB主控建立共享内存并启动数据库,设置系统监控;业务操作数据库,插入、更新或删除数据,并更新到数据区;系统定期从同步区拉取数据,生成持久化文件,并计算文件的SHA-256哈希值;利用事务消息技术,将同步区数据打包成消息,通过自定义协议进行跨中心数据同步;数据接收线程解析数据并按序号存储到红黑树,合并线程将连续的数据块准备好后交由入库线程;通过比较文件名和哈希值实现数据一致性检查,并在发现数据丢失时快速定位和重传数据文件。本发明不仅解决了传统数据同步流程中的瓶颈问题,还为计费业务处理的内存数据库提供了一种新的数据同步技术路径。



1. 面向计费多中心内存数据库的数据同步方法,其特征在于,包括:

步骤S1: ZMDB主控在系统启动时建立共享内存并启动数据库,同时设置系统监控以跟踪数据库关键指标和数据同步服务状态;

步骤S2: 业务通过DBC接口操作数据库,插入、更新或删除数据,并将这些变更序列化到同步区,同时更新到数据区;

步骤S3: 系统定期从同步区拉取数据,按照命名规则生成持久化文件,并计算文件的SHA-256哈希值以便于数据稽核;

步骤S4: 利用事务消息化技术,将同步区数据打包成消息,通过自定义协议进行跨中心数据同步;事务消息化技术具体过程如下:

在业务提交阶段,首先对变更数据操作进行序列化处理,确保数据同步、入库、稽核流程中使用的数据格式一致性;

传输线程在数据传输时会在序列化数据之上组装并发事务协议,准备数据以进行后续的同步操作;

通过多线程并发传输,保障数据同步性能,遇到传输效率不足或链接中断的情况下,选择丢弃当前消息以保护业务操作的流畅性;

同步完成的数据消息通过数据合并检测技术实现快速入库,同时提供异常数据的检测与修复,以保证数据的完整性和一致性;

将接收到的数据消息按序号存入红黑树中,确保数据有序性,然后构造等价的持久化文件内容,并进行数据稽核和跳号处理;

对持久化文件进行稽核,通过哈希值匹配确保数据一致性,在检测到数据缺失时,进行重传流程以修复数据;

分析事务数据,构建前驱事务关系,并进行事务分发入库,优化并发性能,减少数据入库的时延;

步骤S5: 数据接收线程解析数据并按序号存储到红黑树,合并线程将连续的数据块准备好后交由入库线程,确保事务的连续性和数据库的并发性能;

步骤S6: 通过比较文件名和哈希值实现数据一致性检查,并在发现数据丢失时快速定位和重传数据文件以修复数据。

2. 根据权利要求1所述的面向计费多中心内存数据库的数据同步方法,其特征在于,步骤S2具体如下:

步骤A1: 业务系统通过DBC接口执行内存数据库的插入、更新或删除操作;

步骤A2: 执行事务提交,确保通过DBC接口操作变更的数据得到持久化;

步骤A3: 数据访问代理在业务提交期间,将变更数据序列化并优先插入同步区,保证数据的及时同步;

步骤A4: 数据访问代理同时将变更数据更新到业务可查询的数据区,并保持事务的原子性。

3. 根据权利要求1所述的面向计费多中心内存数据库的数据同步方法,其特征在于,步骤S3包括:

步骤B1: 持久化服务定期从同步区拉取数据,并按顺序存储在临时缓冲区中;

步骤B2: 当缓冲区满或遇到特定标记时,将缓冲区数据写入持久化文件,并按照日期

和序号进行命名,同时计算文件的SHA-256哈希值。

4. 根据权利要求1所述的面向计费多中心内存数据库的数据同步方法,其特征在于,步骤S5包括:

步骤D1: 数据接收线程将收到的数据按照协议解析后,按序号存入合并线程的红黑树中,以确保数据的有序性;

步骤D2: 合并线程实时扫描红黑树,将连续的数据组装成数据块,准备进行数据入库操作;

步骤D3: 入库线程将数据块中的数据进行解析,并根据表模型和字段进行高效的批量数据操作。

5. 根据权利要求1所述的面向计费多中心内存数据库的数据同步方法,其特征在于,步骤S6包括:

步骤E1: 在数据稽核阶段,通过比较文件名和哈希值,快速核对数据一致性,并在发现问题时准备数据重传;

步骤E2: 在异常数据修复阶段,通过分析文件名中的日期和序号,快速定位丢失的数据,并进行针对性的数据重传和修复。

6. 面向计费多中心内存数据库的数据同步方法的系统,适用于权利要求1-5中任一项所述的向计费多中心内存数据库的数据同步方法,其特征在于,包括ZMDB数据代理模块、数据同步管理模块和数据持久化模块。

7. 根据权利要求6所述的面向计费多中心内存数据库的数据同步方法的系统,其特征在于,ZMDB数据代理模块包括:

数据访问层:作为业务操作的入口,处理增、删、改、查请求,并在事务提交前将变更序列化到同步区;

数据传输层:从同步区提取序列化数据,通过自定义协议将数据打包并分发到其他中心的同步模块。

8. 根据权利要求6所述的面向计费多中心内存数据库的数据同步方法的系统,其特征在于,数据持久化模块包括:

轮询同步区数据,根据事务顺序和文件标识生成持久化文件,持久化文件附带SHA-256哈希值,用于跨中心的数据一致性比对;

提供快速数据恢复能力以应对数据缺失情况。

9. 根据权利要求6所述的面向计费多中心内存数据库的数据同步方法的系统,其特征在于,数据同步管理模块包括:

数据同步子模块:接收和存储数据同步消息,按数据序号合并成有序数据块,确保同步数据的完整性;

数据入库子模块:解析同步数据,建立前驱事务关系,优化入库性能;

数据稽核子模块:在跳号超时发生时触发,执行数据重传和校验,保障数据完整性和正确性。

面向计费多中心内存数据库的数据同步方法及系统

技术领域

[0001] 本发明涉及数据同步及管理技术领域,具体涉及面向计费多中心内存数据库的数据同步方法及系统。

背景技术

[0002] 在现代运营商计费系统中,随着业务量的剧增及多中心架构的实施,系统内部处理的消息数量迅速上升。特别是在出账、话单处理以及错单回收等关键业务流程中,数据变更的频率和数量常常在短时间内达到巨大的规模。这种激增的数据处理需求往往超出了现有数据同步技术的性能限制,导致在系统遭遇故障时,数据中心可能面临数据丢失的风险。

[0003] 传统的内存数据库同步方法依赖于生成持久化文件,然后对这些文件进行解析和入库处理。然而,在高密度数据处理场景下,数据同步流程中的磁盘I/O操作成为性能瓶颈,严重影响了数据同步的效率,并对系统多中心的高可用性构成威胁。受限于此,当前技术难以满足计费系统中业务的高并发处理需求,尤其是在每月处理上百亿计的话单时,要求系统实现几十万TPS的性能标准。

[0004] 为了保障关键数据的完整性,需要进行严格的数据稽核,并且确保数据同步的及时性以满足恢复点目标(RPO)。此外,计费系统的数据入库效率也对系统的恢复时间目标(RTO)有直接影响,尤其在灾难恢复场景中,需要迅速恢复到最新的数据状态以保证业务的连续性和准确性。在这样的背景下,急需一种新的数据同步方法,以应对高数据密度环境下的多中心数据同步挑战。

发明内容

[0005] 为克服现有技术的不足,本发明提出面向计费多中心内存数据库的数据同步方法及系统,不仅解决了传统数据同步流程中的瓶颈问题,还针对并行性、一致性、可靠性和性能等关键问题提出了创新的解决思路,为计费业务处理的内存数据库提供了一种新的数据同步技术路径。

[0006] 为实现上述目的,本发明提供面向计费多中心内存数据库的数据同步方法,包括:

[0007] 步骤S1: ZMDB主控在系统启动时建立共享内存并启动数据库,同时设置系统监控以跟踪数据库关键指标和数据同步服务状态;

[0008] 步骤S2: 业务通过DBC接口操作数据库,插入、更新或删除数据,并将这些变更序列化到同步区同时更新到数据区,确保事务的原子性和数据的即时同步;

[0009] 步骤S3: 系统定期从同步区拉取数据,按照特定命名规则生成持久化文件,并计算文件的SHA-256哈希值以便于数据稽核;

[0010] 步骤S4: 利用事务消息技术,将同步区数据打包成消息,通过自定义协议进行高效的跨中心数据同步;

[0011] 步骤S5: 数据接收线程解析数据并按序号存储到红黑树,合并线程将连续的数据块准备好后交由入库线程,确保事务的连续性和数据库的并发性能;

[0012] 步骤S6:通过比较文件名和哈希值实现数据一致性检查,并在发现数据丢失时快速定位和重传数据文件以修复数据。

[0013] 进一步地,步骤S2具体如下:

[0014] 步骤A1: 业务系统通过DBC接口执行内存数据库的插入、更新或删除操作;

[0015] 步骤A2: 执行事务提交,确保通过DBC接口操作变更的数据得到持久化;

[0016] 步骤A3: 数据访问代理在业务提交期间,将变更数据序列化并优先插入同步区,保证数据的及时同步;

[0017] 步骤A4: 数据访问代理同时将变更数据更新到业务可查询的数据区,并保持事务的原子性。

[0018] 进一步地,步骤S3包括:

[0019] 步骤B1: 持久化服务定期从同步区拉取数据,并按顺序存储在临时缓冲区中;

[0020] 步骤B2: 当缓冲区满或遇到特定标记时,将缓冲区数据写入持久化文件,并按照日期和序号进行命名,同时计算文件的SHA-256哈希值。

[0021] 进一步地,步骤S5包括:

[0022] 步骤D1: 数据接收线程将收到的数据按照协议解析后,按序号存入合并线程的红黑树中,以确保数据的有序性;

[0023] 步骤D2: 合并线程实时扫描红黑树,将连续的数据组装成数据块,准备进行数据入库操作;

[0024] 步骤D3: 入库线程将数据块中的数据进行解析,并根据表模型和字段进行高效的批量数据操作。

[0025] 进一步地,步骤S4中事务消息技术具体过程如下:

[0026] 数据操作序列化:在业务提交阶段,首先对变更数据操作进行序列化处理,确保数据同步、入库、稽核流程中使用的数据格式一致性;

[0027] 事务传输协议组装:传输线程在数据传输时会在序列化数据之上组装并发事务协议,准备数据以进行后续的同步操作;

[0028] 数据分发传输:通过多线程并发传输,保障数据同步性能,遇到传输效率不足或链接中断的情况下,选择丢弃当前消息以保护业务操作的流畅性;

[0029] 消息合并检测:同步完成的数据消息通过数据合并检测技术实现快速入库,同时提供异常数据的检测与修复,以保证数据的完整性和一致性;

[0030] 数据合并流程:将接收到的数据消息按序号存入红黑树中,确保数据有序性,然后构造等价的持久化文件内容,并进行数据稽核和跳号处理;

[0031] 数据稽核与重传:对持久化文件进行稽核,通过哈希值匹配确保数据一致性,在检测到数据缺失时,进行重传流程以修复数据;

[0032] 前驱事务标定与分发:分析事务数据,构建前驱事务关系,并进行事务分发入库,优化并发性能,并减少数据入库的时延。

[0033] 进一步地,步骤S6包括:

[0034] 步骤E1: 在数据稽核阶段,通过比较文件名和哈希值,快速核对数据一致性,并在发现问题时准备数据重传。

[0035] 步骤E2: 在异常数据修复阶段,通过分析文件名中的日期和序号,快速定位丢失

的数据,并进行针对性的数据重传和修复。

[0036] 面向计费多中心内存数据库的数据同步方法的系统,适用于所述的向计费多中心内存数据库的数据同步方法,包括ZMDB数据代理模块、数据同步管理模块和数据持久化模块。

[0037] 进一步地,ZMDB数据代理模块包括:

[0038] 数据访问层:作为业务操作的入口,处理增、删、改、查请求,并在事务提交前将变更序列化到同步区;

[0039] 数据传输层:从同步区提取序列化数据,通过自定义协议将数据打包并分发到其他中心的同步模块。

[0040] 进一步地,数据持久化模块包括:

[0041] 轮询同步区数据,根据事务顺序和文件标识生成持久化文件,持久化文件附带SHA-256哈希值,用于跨中心的数据一致性比对;

[0042] 提供快速数据恢复能力以应对数据缺失情况。

[0043] 进一步地,数据同步管理模块包括:

[0044] 数据同步子模块:接收和存储数据同步消息,按数据序号合并成有序数据块,确保同步数据的完整性;

[0045] 数据入库子模块:解析同步数据,建立前驱事务关系,优化入库性能;

[0046] 数据稽核子模块:在跳号超时发生时触发,执行数据重传和校验,保障数据完整性和正确性。

[0047] 与现有技术相比,本发明的有益效果是:

[0048] 1.本发明提供了面向计费多中心内存数据库的数据同步方法及系统,将传统基于持久化文件的数据同步过程,改进为基于消息的数据消息备份和数据消息入库流程,提高了数据处理的效率;通过序列化和事务号生成机制,确保了数据在同步和入库过程中的完整性和有序性,避免了数据冲突和不一致的情况。

[0049] 2.本发明提供了面向计费多中心内存数据库的数据同步方法及系统,通过多线程并发传输和前驱事务标定分发技术,显著提高了数据同步和入库的并发处理能力,减少了入库时延。

[0050] 3.本发明提供了面向计费多中心内存数据库的数据同步方法及系统,实现了消息合并检测技术,确保即使在高并发无序数据流中,也能通过红黑树结构和持久化文件哈希值校验,保持数据的完整性和准确性。

[0051] 4.本发明提供了面向计费多中心内存数据库的数据同步方法及系统,对于数据传输中可能遇到的效率下降或连接中断问题,采取了消息丢弃策略,避免了业务操作的同步区阻塞,同时通过数据稽核和重传流程,强化了数据修复和异常恢复能力。

[0052] 5.本发明提供了面向计费多中心内存数据库的数据同步方法及系统,构建了一套闭环的多中心数据同步方案,即使在压力场景下也能保障核心数据不丢失,大幅提升了多中心系统的可用性和健壮性。

[0053] 6.本发明提供了面向计费多中心内存数据库的数据同步方法及系统,以恢复点目标(RPO)为主要目标,在数据备份阶段确保数据不丢失;以恢复时间目标(RTO)为次要目标,在数据入库阶段优化处理速度。

附图说明

[0054] 为了更清楚地说明本发明具体实施方式或现有技术中的技术方案,下面将对具体实施方式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,下面描述中的附图是本发明的一些实施方式,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。

- [0055] 图1是本发明系统流程示意图;
- [0056] 图2是本发明系统架构示意图;
- [0057] 图3为本发明序列化格式示意图;
- [0058] 图4是本发明数据序列化格式示意图;
- [0059] 图5是本发明序列化数据上组装并发事务协议示意图;
- [0060] 图6(a)是本发明数据同步任务分发流程;
- [0061] 图6(b)是本发明数据传输线程工作流程;
- [0062] 图7是本发明数据合并架构示意图;
- [0063] 图8是本发明数据流程架构示意图;
- [0064] 图9是本发明前驱事务构建流程示意图。

具体实施方式

[0065] 下面将结合附图、通过对本发明的优选实施方式的描述,更加清楚、完整地阐述本发明的技术方案。

[0066] 术语解释:

[0067] MDB(Memory Database,内存数据库):基于共享内存构建的关系型数据库系统,由于核心系统脱离磁盘依赖,访问性能基本不受I/O影响,一般用于加速高性能要求的业务系统。

[0068] ZMDB(ZsmartMemory Database):分布式关系型内存数据库,主要用于存储计费高频访问的核心数据,加快计费应用的数据访问性能。

[0069] DBC(Database Connectivity,数据库连接):涉及到在应用程序与数据库之间建立、管理、使用连接的各种标准,进行数据同步时要使用DBC确保事务操作正确性。

[0070] 数据持久化:ZMDB作为内存数据库需要定时定量将内存变更数据同步到磁盘数据库中,避免主机/内存异常时,核心数据丢失,将内存中变更数据持续同步到磁盘库中的操作称为数据持久化。

[0071] 持久化文件:ZMDB进行数据持久化的媒介物,存储业务表数据序列化后的二进制数据,与MYSQL的REDO日志作用相同。

[0072] 数据同步:指数据以一定的媒介(比如按照消息、文件等),传输到其他数据中心,不一定能立即提供业务访问。

[0073] 数据入库:指将数据通过DBC接口操作到其他数据库中,并能正常提供业务访问。

[0074] 事务提交:在数据库事务处理中,将所有已执行的操作(如插入、更新、删除数据)永久保存到数据库中的过程。

[0075] 持久化同步(内存数据库):为了避免对实时业务产生阻塞影响,以内存数据库的持久化文件为基础,包含持久化文件生成流程、持久化文件解析流程,及DBC接口调用入库

流程。

[0076] RPO (Recovery Point Objective, 恢复点目标):指在多中心高可用性系统中,当一中心发生故障时,系统或应用可以接受的数据丢失量。在本例中,该指标通常关注数据备份率,负责标定计费系统高可用能力。

[0077] 数据消息备份:指在计费多中心高可用系统中,以保障RPO指标的数据同步操作,即关心数据不丢失,而非业务数据访问能力。

[0078] RT0 (Recovery Time Objective, 恢复时间目标):指在多中心高可用性系统中,当一中心发生故障时,系统或应用必须在预定的时间内恢复到正常状态的目标时间。在本例中,该指标通常关注数据同步完成时间,负责标定计费系统高可用能力。

[0079] 数据消息入库:指在计费多中心高可用系统中,为了满足计费准确性要求,以保障RT0指标的数据入库操作,即业务对准确数据访问能力。

[0080] TPS (Transactions Per Second):每秒处理事务数量,负责标定计费系统业务处理能力。

[0081] 如图1所示,本发明具体为:

[0082] 1. ZMDB主控负责创建共享内存并启动内存数据库,多个中心的不同ZMDB实例都需要通过主控服务启动,主要流程如下:

[0083] 1.1步骤M1为ZMDB启动流程,系统初次启动时构建包括同步区(用于记录业务过程操作)、数据区(用于存储业务访问数据)等共享内存结构,并启动数据代理服务、持久化服务、数据同步服务。数据同步服务启动后,自动连接其他中心的数据同步服务。

[0084] 1.2步骤M2为系统监控通知流程,除了数据库关键指标外,还对数据同步关键服务和数据同步进度进行监控。

[0085] 2. ZMDB提供标准DBC接口供计费业务使用,业务操作数据及提交后,数据会自动写入同步区,具体流程如下:

[0086] 2.1步骤A1为业务通过DBC接口操作内存数据库的流程,当且仅当对数据进行插入、更新或删除操作时,才需要通过事务提交让变更数据入库。

[0087] 2.2步骤A2为业务通过DBC接口提交步骤A1中操作变更的数据。

[0088] 2.3步骤A3为数据访问代理在业务提交期间,优先将变更数据及操作序列化,并插入同步区确保数据同步及时率。同步区为循环队列,插入同步区操作全局互斥,通过数据序号单调递增,保证数据准确性和有序性。

[0089] 2.4步骤A4为数据访问代理在业务提交期间,将变更数据修改到业务可正常访问的数据区。步骤A3、A4在业务提交操作中一次性完成,并符合事务原子性要求。

[0090] 2.5步骤B1为拉取同步区数据操作。拉取操作严格有序,取下来的数据会根据持久化文件名,存入对应的buffer中。

[0091] 2.6步骤B2为生成持久化文件。当遇到文件尾数据时,将buffer中的数据落地生成持久化文件。持久化文件命名以“redo_<日期(yyyymmddHHMMSS)>_<序号(xxx)>_<后缀>”的方式展开,同一秒内生成多个文件时序号从‘000’递增,从而保证根据文件名就可以圈定数据执行顺序和数据范围。每个持久化文件会根据文件内容计算SHA-256哈希值,在进行数据稽核时只需要比较文件名及哈希值,就能确定数据是否一致,保障数据完整性和准确性。

[0092] 3. ZMDB通过事务消息化传输技术,实现脱离磁盘I/O且高实时率的数据同步,同步

流程不阻塞业务提交操作,主要过程如下:

[0093] 1) 步骤C1为拉取同步区数据操作。与步骤B1类似,但取下来的数据会以事务为维度进行组装分发,确保数据同步过程事务完整性。分发过程轮询传输线程存储队列实现,入队及出队操作需要加锁,确保队列操作的准确性。入队时以事务为维度,出队时以队列事务个数为维度批量出队,从而减少分发流程和传输流程的冲突率,提升并发性能。

[0094] 2) 步骤C2为多线程传输操作。实时扫描本队列已存储事务个数,当事务个数大于0时,进行加锁批量出队。多个事务通过批量绑定的方式,以单个消息进行传输,降低网络I/O多次调用的性能损耗。考虑传输协议的效率和复杂度,选择自定义协议进行多事务消息化传输。各个线程之间传输过程互不影响,保障数据同步过程性能。当数据传输到其他中心后,就已经完成数据消息备份,虽然此时数据还未真实入库,但已经能保障通过该数据消息完成数据恢复点的刷新,即系统RPO指标。通过屏蔽磁盘I/O(完全内存化)、降低网络I/O(批量绑定转发)、优化数据传输格式及内容(数据消息序列化),保证了数据同步的及时率。

[0095] 4. ZMDB通过前驱事务标定分发技术,实现关联数据串行处理,非关联数据并行处理,且保障处理线程互不阻塞,主要过程如下:

[0096] 1) 步骤D1为数据接收线程解析汇聚流程。数据同步模块的数据接收线程为多线程,与原中心的数据传输线程一一匹配。当接收到待处理的数据后,将数据按协议中进行解析,并根据数据需要将数据存储合并线程的红黑树中,红黑树以序号计算存储位置,从而确保从红黑树中读取数据严格有序。数据同步模块的合并线程为单线程,实时扫描红黑树并拉取树中节点信息,将连续数据组装成数据块。该数据块以持久化文件为维度,与原中心B2流程生成的持久化文件内容完全一致,因此可以使用相同的算法计算哈希值进行校验。当数据块中的数据完整时,就可以摘除红黑树中已处理的节点,保障系统内存占用可控。处理完成的数据块用于传递给数据入库线程,保障数据消息入库的及时性。

[0097] 2) 步骤D2为前驱事务解析及事务分发流程。合并线程完成数据合并后,将可以处理的数据进行事务解析,将关联事务(操作相同表相同主键的数据)和非关联事务进行拆分,确保数据库入库的并发性能。对于关联事务,根据数据顺序构建前驱事务,保障关联事务处理的有序性(例如先删除、再插入的相同主键数据,不能以先插入再删除入库)。前驱事务关系中,相同更新操作、更新字段、更新主键的连续数据,进行事务合并操作,以最后一次操作为基准,减少相同数据反复更新的性能损耗(例如对同一个账户充值10笔,每次充值10元,最后一次该账户余额为100元,数据入库时直接将该账户余额更新为100元即可)。前驱事务构建完毕后,以前驱事务为分发原则,将事务分配到各个入库线程中,既保障事务处理顺序,又保障了入库并发性能。

[0098] 3) 步骤D3为数据库入库操作流程。入库线程以表模型、更新字段为依据,构建不同的数据操作对象,确保数据操作对象的可复用性,实现数据操作句柄缓存,提升数据操作性能。同时,在相同数据操作句柄上,采用批量操作、批量提交的方式,降低数据入库的性能损耗,提升数据入库及时性。

[0099] 5. ZMDB通过消息合并检测技术,实现异常数据的发现与修复,确保计费业务核心数据同步的准确性,主要过程如下:

[0100] 1) 步骤E1为数据稽核重传流程。在数据稽核阶段,通过传递文件名和哈希值,与原中心相同持久化文件名的数据哈希进行比较,实现数据一致性快速稽核。在数据重传阶段,

通过传递跳号前后的文件名,根据文件名中携带的日期及序号,快速圈定重传修复范围,再以文件维度传回快速修复数据。数据序号跳号一般发生在网络异常、主机检修、服务重启的情况下,实现自动化的多中心数据快速修复效果。

[0101] 2) 步骤E2为异常数据文件检索流程。通过异常数据中心传递的跳号前后文件名,在原中心以日期为范围快速检索丢失数据。在A3阶段已经确定文件名、数据序号、事务号之间的关系,且在后续流程中不进行变更,因此可以通过重传文件的方式,修复丢失的数据。由于检索维度以文件为准,不捞取文件中丢失的数据序号,也提升了数据检索修复的及时率。

[0102] 本申请方案涉及到多个模块和处理流程,这个方法的实施步骤可以概括为以下几个阶段:

[0103] 1. 系统初始化:

[0104] 1) 启动ZMDB实例,创建共享内存数据库。

[0105] 2) 启动数据访问代理进程、持久化进程及数据同步管理进程。

[0106] 3) 多个不同中心的ZMDB实例相互注册,构建多中心数据中心,其中数据传输层向个数据中心的数据管理模块登陆注册,建立连接。

[0107] 4) 批量启动计费某业务模块,为了确保话单处理不重复,一般将相同业务指向其中某一个数据中心的ZMDB实例。

[0108] 2. 业务处理:

[0109] 1) 在进行业务处理后,调用ZMDB提供的DBC接口,将增、删、改数据操作发给ZMDB数据代理。

[0110] 2) ZMDB数据代理收到操作请求后,直接连接本地共享内存,按照ZMDB内部数据处理流程,对业务数据进行对应操作。

[0111] 3) 业务完成逻辑处理后,申请提交操作。数据提交阶段优先对业务操作的事务内所有数据进行序列化操作,按照业务操作顺序插入同步区中。其中,根据全局序列记录序号信息,保证数据处理严格有序;根据定时定量策略记录持久化文件名、文件首、文件尾,保证数据稽核、重传性能。同时,存储的业务数据按照一定格式进行组装,既实现对增、删、改不同的操作进行描述,又能减少无效数据的传输,提升后续数据同步及入库性能。

[0112] 4) 当且仅当提交操作完成后,变更数据才能被永久保存入内存数据库中。

[0113] 3. 数据持久化:

[0114] 1) 数据持久化实时扫描流程按照单线程运行。从同步区中有序扫描出入队完成的数据,并根据数据包含的持久化文件名、文件首、文件尾标签,生成持久化文件。

[0115] 2) 根据持久化文件中内容,提前计算SHA-256哈希值,用于数据一致性稽核。

[0116] 4. 数据同步:

[0117] 1) 数据传输层实时扫描流程按照单线程运行。从同步区中有序扫描出入队完成的数据,以事务为单位分配给数据传输线程。

[0118] 2) 数据传输层消息传输流程按照多线程运行。对分配完成的数据按照自定义协议打包,远程传递给其他中心的数据同步模块,该过程并发无阻塞,保障了数据同步性能。传输同样以事务为单位,单个事务一般包含多条记录,批量传输时可以大幅优化网络I/O的性能损耗。

[0119] 3) 数据同步消息接收流程按照多线程运行。收到消息后,反序列化得到数据序号,并以数据序号为维度,将数据存储于红黑树构造的存储单元中。同时,在响应包中反馈已经合并完成的持久化文件名。

[0120] 4) 数据同步数据合并流程按照单线程运行。将存储单元中的数据取出,并以数据序号为依据,对数据进行排序处理,保证数据不丢失。同时,反序列化得到数据序号、事务号、持久化文件名、文件首、文件尾及数据块等关键信息。以持久化文件名、文件首、文件尾为依据,将相同文件下的数据有序组装成与生成的持久化文件相同的数据块,该数据块一般包含多个事务。当遇到数据序号不连续的场景时,以跳号超时时间为依据,记录跳号前后的两个数据对应的持久化文件名。

[0121] 5. 数据入库:

[0122] 1) 事务解析流程按照单线程运行。当存在已合并完成的事务时,按照合并顺序对事务展开解析。解析事务中包含表、操作、数据信息,其中表信息包括本次操作同步的表名、表字段;操作信息包括本次操作同步的操作类型,具体为插入(insert)、更新(update)、删除(delete);数据信息包括本次操作同步的数据内容。数据内容根据数据操作类型存储发生变更,当操作类型为插入时,包含该表所有字段的数据;当操作类型为更新时,包含该表更新字段及主键数据;当操作为删除时,仅包含该表主键数据。通过区分场景的数据封装动作,大幅降低每次数据同步传输的大小,保障了数据同步性能。

[0123] 2) 事务解析流程根据事务中包含的表、操作、数据信息,为需要入库的数据建立前驱事务关系。具体步骤为,根据操作表、操作主键计算哈希,对哈希值相同的数据操作建立前驱事务关系。前驱事务中,对于更新操作、连续且相同更新字段的数据进行合并,减少冗余数更新,提升数据入库性能。根据建立好的前驱事务关系,将事务数据分配到不同的数据库操作线程中,对于存在前驱事务关系的数据优先分配到相同线程中,对于不存在前驱事务关系的数据,根据处理线程数据积压情况优先分配到空闲线程。

[0124] 3) 数据库操作流程按照多线程运行。前驱事务关系确保了相同行级别操作的数据优先分配到相同线程,减少不同线程操作相同数据的冲突率,保障数据入库并发性能。

[0125] 6. 数据稽核:

[0126] 1) 触发数据跳号重传流程时,以数据同步记录的跳号前后持久化文件为依据,向原中心请求数据重传。数据重传以跳号前的持久化文件名为起始文件,以跳号后的持久化文件名为结束文件,传递包含起始、终止文件在内的范围内所有文件。当原中心文件重传完毕后,将数据同步数据合并流程中的所有覆盖数据全部过滤。同时,遍历至下一个非过滤数据,恢复正常合并流程。该数据一定是不同持久化文件名的文件首,且数据序号与重传终止文件中的最后一个数据序号连续不中断。

[0127] 2) 每一个合并完成的数据块,会根据数据块内容计算SHA-256哈希值,并与原中心相同持久化文件名的文件哈希值进行比较。当哈希值完全相同时,标识数据准确一致,反之该文件需要重传修复,从而保障数据准确性。

[0128] 面向计费多中心内存数据库的数据同步方法的系统,如图2所示,本例中,由于ZMDB系统模块较为复杂,大部分模块和数据同步未有直接关系,故只抽取与数据同步有关的部分进行架构描述。

[0129] 1. ZMDB数据代理

[0130] ZMDB数据代理作为ZMDB数据访问的对外入口,实现业务对ZMDB数据的增、删、改、查及事务操作。

[0131] 1) 数据访问层

[0132] 数据访问层负责ZMDB数据增、删、改、查及事务操作的载体。本例中,需要保障数据在正式提交前就完成数据同步操作,在事务提交阶段优先把数据序列化后放入同步区中。同步区为基于共享内存的循环队列,构造用于持久化及数据同步时的出队偏移。数据序列化涉及将业务数据格式化、并补充业务数据涉及的序号、事务号、持久化文件名、文件头标识、文件尾标识、操作、表、分片、索引等信息,既用于数据一致性稽核校验,也用于后续反序列化进行数据入库操作。其中,序列化产生的持久化文件名,会严格按照时间标签进行区分,作为数据稽核重传的关键标识。而涉及的文件头标识、文件尾标识,则按照定时定量的原则打标,有助于减少数据持久化的I/O损耗。

[0133] 2) 数据传输层

[0134] 数据传输层负责将同步区中的数据按事务捞取,快速分发到传输线程中。传输线程以自定义协议的方式,与其他中心的数据同步管理模块进行交互,发送分配的数据块。各传输线程互相独立,传输过程并行无序,从而保障传输并发性能。同时,数据传输协议还负责获取其他中心数据同步进度,及时通知各中心数据块遗漏的场景。

[0135] 2. 数据持久化

[0136] 数据持久化负责轮询将当前中心同步区数据取出,根据文件头标识、文件尾标识落地生成持久化文件。持久化文件生成时严格遵循事务执行顺序,与数据同步操作数据块也完全一致。本例中,持久化文件会生成各自的SHA-256哈希值,通过比较其他中心数据同步生成的SHA-256哈希值,来标定文件生成的一致性。此外,在数据缺失的场景下,也可以使用原中心生成的持久化文件直接覆盖,大幅优化了海量数据稽核、修复困难的问题。

[0137] 3. 数据同步管理

[0138] 本例中面对的计费多中心架构,实际为多中心多活架构,非冷热部署架构。因此,无论当前中心还是其他中心,都具备数据同步模块。

[0139] 当业务数据产生在当前中心时,其他中心数据同步模块主要承接数据同步、数据入库及数据检测功能,而当前中心数据同步模块主要承接数据检测功能,反之亦然。由于整个数据同步、入库流程完全不依赖持久化文件,大幅优化磁盘I/O对数据同步、数据入库的性能影响,确保了数据同步、入库的及时性。

[0140] 1) 数据同步

[0141] 数据同步模块主要包含数据传输接收流程和数据合并流程。数据传输接收流程采用多线程并行的方式,接收数据同步消息,并将数据同步消息以数据序号维度进行存储和传递;数据合并流程则采用单线程运行的方式,根据合并规则将数个无序消息合并成与原中心持久化文件内容相同的数据块,并严格保证数据有序性。当且仅当数据序号跳号超时,会将跳号序号作为数据传输的响应报文,供数据检测环节进行跳号检测。

[0142] 2) 数据入库

[0143] 数据入库模块主要包括事务解析流程和数据库操作流程。事务解析流程采用单线程运行的方式,通过反序列化同步数据的方式,获取数据及关键信息。同时,事务解析流程会将数据根据表、操作、数据信息建立哈希表,以事务为单位建立前驱事务关系,确保数据

入库的严格顺序。在建立哈希表时和前驱事务关系时,会自动合并相同维度下的更新数据操作,考虑到计费业务中关键数据(例如余额、累积量)大多以更新为准,能大幅优化相同用户/账户入库数据操作频次,从而大幅提升数据入库性能。当事务解析完将数据分配到多个数据库操作线程中,具备前驱事务关系的数据优先放到同一个线程中处理,避免线程互相阻塞等待。

[0144] 3) 数据稽核

[0145] 当在数据同步过程中触发跳号超时,由数据稽核模块完成数据重传及校验的流程。实际生产中,数据跳号重传一般发生在中心重启或网络异常丢包的情况下,属于异常恢复流程。考虑到数据跳号的复杂性(不一定是只跳一个),以及海量数据中定位遗漏数据性能较慢,故采用持久化文件名分段的策略进行数据重传。将数据序列化时包含的持久化文件名作为遗漏数据范围判断的依据,通过记录当前最后一个正常数据的持久化文件名,及下一个跳号数据的持久化文件名,能准确判断出跳号范围中涉及的所有数据。最终,以持久化文件为维度,一次性传输包含遗漏数据的所有文件内容,并在数据同步流程中过滤相同数据。该流程确保了计费系统数据的安全可靠。

[0146] 作为一种具体的实施例,事务消息传输技术具体如下:

[0147] 对于传统内存数据库数据同步而言,一般采用文件作为媒介物,并在事务提交完成后才开始数据同步处理,因此磁盘I/O、网络I/O、处理流程均会影响数据同步及时性,反映在系统RPO指标上时,很难在本中心业务性能和多中心数据恢复点上取得平衡。特别是在业务高并发场景下,并发操作内存数据库数据变更的性能已经远超磁盘I/O的性能,在事务提交阶段进行同步文件落地会严重遏制业务性能,导致业务无法及时处理完成。而在事务提交完成后,再进行同步文件落地,则导致有数分钟的文件无法及时同步到其他中心,此时本中心异常会导致数据遗失数分钟,造成重大损失。同时,若对本中心数据操作不选择落地的方式,一旦同步过程出现断网、主机异常、服务异常等场景后,数据也会丢失,数据准确性无法保障。

[0148] 事务消息传输技术,是将本地变更数据存储与远程多中心数据消息同步进行解耦,将数据以事务的维度进行消息传输,专注数据消息备份的技术。系统内部,将事务消息流程提前到事务提交中阶段,在事务数据正式入库前,就将数据消息存入同步区中进行传输。同时,事务消息通过大幅压缩数据维度,以数据不同操作为准,决定数据传输内容,实现真实同步的数据大小远小于本地数据入库的数据大小。而同步区中相同的数据,也被复用于持久化流程,与数据同步传输流程互不干扰。在上述方案下,一方面提升的数据同步流程对业务操作性能不产生影响,保障业务核心处理逻辑的稳定性;另一方面数据同步流程完全脱离持久化流程,脱离磁盘I/O,让数据同步的及时率得到保证;最后,本地数据变更的操作日志仍然通过数据持久化保存下来,在内存数据库数据易失的前提下,保障了核心数据的准确性和完整性。

[0149] 数据操作序列化:

[0150] 在业务提交阶段,会优先对变更数据操作进行序列化,确保后续数据同步、数据入库、数据稽核流程使用的数据对象一致性。数据操作序列化是将业务操作数据结合业务操作类型,转化成一定协议格式的二级制数据,其序列化格式如图3所示:

[0151] 1、表名

- [0152] 32字节字符型,业务操作表的名称,满足数据模型规范中表名长度。
- [0153] 2、操作类型
- [0154] 4字节整型,描述业务操作数据的方式,分为插入、更新、删除。
- [0155] 3、数据源类型
- [0156] 4字节整型,描述业务类型,部分业务类型不需要进行数据同步。
- [0157] 4、操作日期
- [0158] 8字节整型,描述业务操作数据时间戳。
- [0159] 5、数据序号
- [0160] 8字节整型,ZMDB系统内部全局序列生成,单调递增,不同数据序号不相同,从而保证数据完整性和有序性。
- [0161] 6、事务号
- [0162] 8字节整型,ZMDB系统内部全局序列生成,以业务提交事务为维度,同事务中所有数据的事务号一致。
- [0163] 7、持久化文件名
- [0164] 32字节字符型,生成持久化文件的名称,文件名生成格式为“redo_<日期(yyyymmddHHMMSS)>_<序号(XXX)>_<后缀>”。通过解析文件名中的日期,可以快去确定文件重传的范围。
- [0165] 8、文件首
- [0166] 1字节字符型,描述持久化文件的第一个数据。持久化文件包含的所有事务均是完整的事务,因此持久化文件的第一个数据,也是该持久化文件中第一个事务的第一个数据。
- [0167] 9、文件尾
- [0168] 1字节字符型,描述持久化文件的最后一个数据。持久化文件包含的所有事务均是完整的事务,因此持久化文件的最后一个数据,也是该持久化文件中最后一个事务的最后一个数据。
- [0169] 10、数据长度
- [0170] 4字节整型,描述该次操作同步的数据长度。
- [0171] 对于不同操作类型而言,数据序列化格式如图4所示:
- [0172] 1、插入操作
- [0173] 描述本次插入操作涉及的所有字段,根据表中字段存储的顺序,存储插入字段数据。
- [0174] 2、更新操作
- [0175] 描述本次更新操作涉及的更新字段及数据主键,屏蔽非更新字段,减少数据传输的大小。
- [0176] 3、删除操作
- [0177] 描述本次删除操作涉及的数据主键,屏蔽非主键字段,减少数据传输的大小。
- [0178] 事务传输协议:
- [0179] 传输线程对数据进行传输时,会在序列化数据上组装并发事务协议,具体如图5所示:
- [0180] 数据分发传输:

[0181] 数据消息备份传输流程,支持多线程并发传输,保障同步性能。当系统传输效率无法满足业务操作性能时,当前消息选择直接丢弃,避免导致业务操作同步区阻塞;当向其他中心同步的链接中断或异常时,当前消息选择直接丢弃,避免中心异常阻塞数据传输。被丢弃的消息后续根据跳号检测及重传流程完成数据修复。由于传输并发性能远高于业务性能,上述异常场景一般发生在中心异常的情况下,因此优先选择保障本服务数据同步流程稳定性。具体流程如图6(a)、图6(b):

[0182] 图6(a)所示,数据同步任务分发流程,通过同步区解耦事务提交操作,实现不影响业务提交流程的基础上,以事务维度分发数据传输任务,在所有传输线程繁忙并超时时(例如网络异常),及时丢弃超时数据,避免同步区满引发的业务提交阻塞场景。

[0183] 图6(b)所示,数据传输线程工作流程,通过自定义协议完成与服务端数据同步传输,并通过心跳/重连的方式保障与服务端连接的稳定性,同时在响应报文中确认当前服务端同步完成的进度,确保传输的及时性和有效性。

[0184] 消息合并检测技术:

[0185] 在数据同步完成后,系统RPO指标实质已经完成,单中心故障后数据能保证不丢失。此时,同步完成的数据消息如何快速入库,提供业务准确访问能力变得至关重要。在数据消息入库阶段,一方面需要考虑数据入库的有序性和准确性,要保障通过高并发同步的数据文件完整可用;另一方面也需要考虑异常场景下的数据恢复机制,提升系统健壮性和鲁棒性。

[0186] 消息合并检测技术,就是在面对高并发、无序、零散数据时,集成解析、组装、检测、修复的一套闭环系统。数据消息按照数据、事务、文件构成三层逻辑结构,首先通过数据序号保障了数据有序性,其次通过事务序号及事务个数保障事务一致性,最后通过文件名及文件首位标记确定数据完整性。同时,系统可以针对性地检索数据顺序异常,事务不一致异常及文件数据缺失异常,并采用文件修复覆盖的机制,把成千上万的数据检测过程简化成文件稽核修复过程,保障了数据恢复流程的性能和可靠性。数据合并如图7所示;

[0187] 数据合并流程是以构造原中心完全等价的持久化文件内容为核心设计,实现多中心在数据同步内容上的完全一致性,在保障数据准确性、有序性的基础上,大幅简化数据稽核、重传修复的流程。在多线程的数据接收流程接收完消息后,会严格互斥的将消息以数据序号为KEY存入红黑树中,此时遍历红黑树的流程可以确保访问数据序号严格有序。当取到完整且连续的序号时,将对应事务数据拼接到以持久化文件名为维度的BUFFER中。对应文件处理完毕时,该BUFFER会进行一次哈希运算,计算SHA-256哈希值,并立即与原中心进行稽核校验。当取到不连续的序号时,会进行跳号等待,跳号超时就通过上下文关联信息进行数据重传。具体流程如图8所示,数据同步合并流程,通过判断序号的连续性,保障并发无序传输的数据同步消息合并成与客户端持久化文件完全一致的数据内容,并对跳号等待超时的场景进行中断,保障数据同步流程数据的准确性;

[0188] 数据稽核:

[0189] 数据稽核主要通过匹配对应文件名的哈希值完成,为独立线程。稽核流程以文件为单位,在一次网络交互中即可完成一份文件数据稽核。当稽核完成哈希值不匹配时,直接传递完整正常文件内容,对异常数据进行覆盖处理。

[0190] 数据重传:

[0191] 重传流程主要包含三个环节:重传触发、补发文件及重传处理。

[0192] 1、[重传触发]所有重传感知均由数据合并流程负责,进入跳号场景后,会在一定时间内匹配数据序号丢失。确定数据丢失时,直接将丢失数据上下文,即丢失数据目标的前一个数据的持久化文件名,丢失数据目标的后一个数据的持久化文件名,传递给原中心的数据管理模块。

[0193] 2、[补发文件]收到重传请求后,数据管理模块解析请求中的持久化文件名,获得重传开始日期、重传结束日期。由于多中心在数据同步内容上强一致,因此对应日期的持久化文件必定存在。通过对本地的持久化文件目录进行扫描,捞出包含开始、结束日期范围内的所有文件后,获取文件内容打包返回。当需要重传的文件跨天时,可以按涉及日期到指定目录下进行扫描,扫描完成后再统一汇总传输。

[0194] 3、[重传处理]当数据中心收到重传响应时,以文件维度清理已完成合并的BUFFER;以事务维度重新对数据入库流程发布事务进行重处理;以数据维度清理红黑树中所有包含的数据,被清理数据实质已经包含在不发文件中,为冗余数据。

[0195] 前驱事务标定分发技术:

[0196] 计费业务中存在大量关联操作数据,通过关系型数据库的事务一致性保障业务处理的准确性,因此在进行数据同步时,仍然需要保障事务操作维度完全一致有效,因此实际入库过程中需要按照事务进行分发处理。由于事务中通常包含大量数据操作,采用轮询分发时各数据库处理线程阻塞情况严重,导致数据入库性能低下。

[0197] 前驱事务标定分发技术提供一种判断事务之间关系的方法,通过判断事务之间是否存在数据互相冲突,标定事务的前后执行顺序,在数据分发入库之前就明确了事务之间关系,保障了数据处理准确性的同时,降低处理现场之间的冲突阻塞。同时,对计费业务中常见的相同数据多次更新的操作,进行了过滤处理,在不破坏事务一致性的前提下,实现了数据库操作频率的降低,保障数据入库性能。

[0198] 前驱事务构建:

[0199] 前驱事务以操作表名、操作类型、操作字段、主键为计算维度,为后续数据处理的并发分布提供依据。当一个事务中包含多条记录时,不同记录的前驱事务可能不同,因此一个事务的前驱事务是一个集合。同理,当前事务也有可能是后续数个不同事务的前驱事务。因此,前驱事务通过数组+哈希表的方式进行描述,并在事务中通过数组进行存储体现。构建前驱事务的过程中,对连续操作相同数据的行为进行过滤,该流程在不对事务一致性进行影响的前提下,大幅减少数据操作频率,保障数据入库性能,具体流程如图9所示,前驱事务标定流程,通过数据操作等维度计算哈希并关联事务执行顺序,同时采用了一种操作合并过滤的方法,既保障数据入库的有序性,又减少需要同步入库的数据量。

[0200] 事务分发入库

[0201] 对指定个数的BUFFER块进行解析,并构建前驱事务后,将事务根据前后关联关系分发到数据库处理线程中。关联事务优先分配到相同线程中处理,根据事务关联关系建立串行处理逻辑;非关联事务优先平均分配到空闲线程中,保障数据入库并发能力。

[0202] 在数据处理线程中,会根据表名、操作类型、操作字段组装操作SQL,不同的操作SQL匹配到不同的DAO(数据访问对象)中,并分配访问句柄。通过缓存访问句柄,减少数据库SQL解析次数及网络I/O损耗,并通过批量绑定进一步减少数据入库时延,提升入库性。

[0203] 作为一种具体的实施例,本发明效果如下:

[0204] 1) 数据备份同步及时性高

[0205] 对标生产系统,构建双中心ZMDB,模拟单中心10、20、40、80万TPS的业务性能,在A中心中进行积压等待后,检查B中心数据同步日志及时率。性能数据如表1所示:

[0206] 表1

	10 万数据	20 万数据	40 万数据	80 万数据
持久化文件数	21	44	92	230
事务传输数	233	381	851	1529
同步及时率	189ms	323ms	672ms	1281ms

[0207] 数据同步及时率主要和网络交互次数及报文大小相关,目前1秒内可完成预计60万左右的数据同步,可以满足计费系统峰值性能要求。

[0209] 2) 数据入库高性能

[0210] 采用与上述相同的验证场景,检测B中心测试数据实际入库情况,性能数据如下,部分数据通过前驱事务构建流程进行合并,如表2所示:

[0211] 表2

	10 万数据	20 万数据	40 万数据	80 万数据
实际同步数据量	93613	173992	379823	773291
入库及时率	731ms	1581ms	2966ms	5871ms

[0212] 数据同步到B中心后,还需要消耗对应时间完成入库,入库性能受限于入库线程数(64),并发性能不超过15万TPS,与计费业务平均性能基本一致,即保证大部分情况下入库不积压。少数压力场景下,会导致入库及时率积压,但基本可以在5分钟内完成积压数据入库。

[0214] 3) 多中心系统高可用

[0215] 采用与上述相同的验证场景,在数据同步、入库的过程中,反复启停B中心业务,保障数据丢失率为0,文件数据修复率达到至少10个文件/秒。

[0216] 上述具体实施方式仅仅对本发明的优选实施方式进行描述,而并非对本发明的保护范围进行限定。在不脱离本发明设计构思和精神范畴的前提下,本领域的普通技术人员根据本发明所提供的文字描述、附图对本发明的技术方案所作出的各种变形、替代和改进,均应属于本发明的保护范畴。本发明的保护范围由权利要求确定。

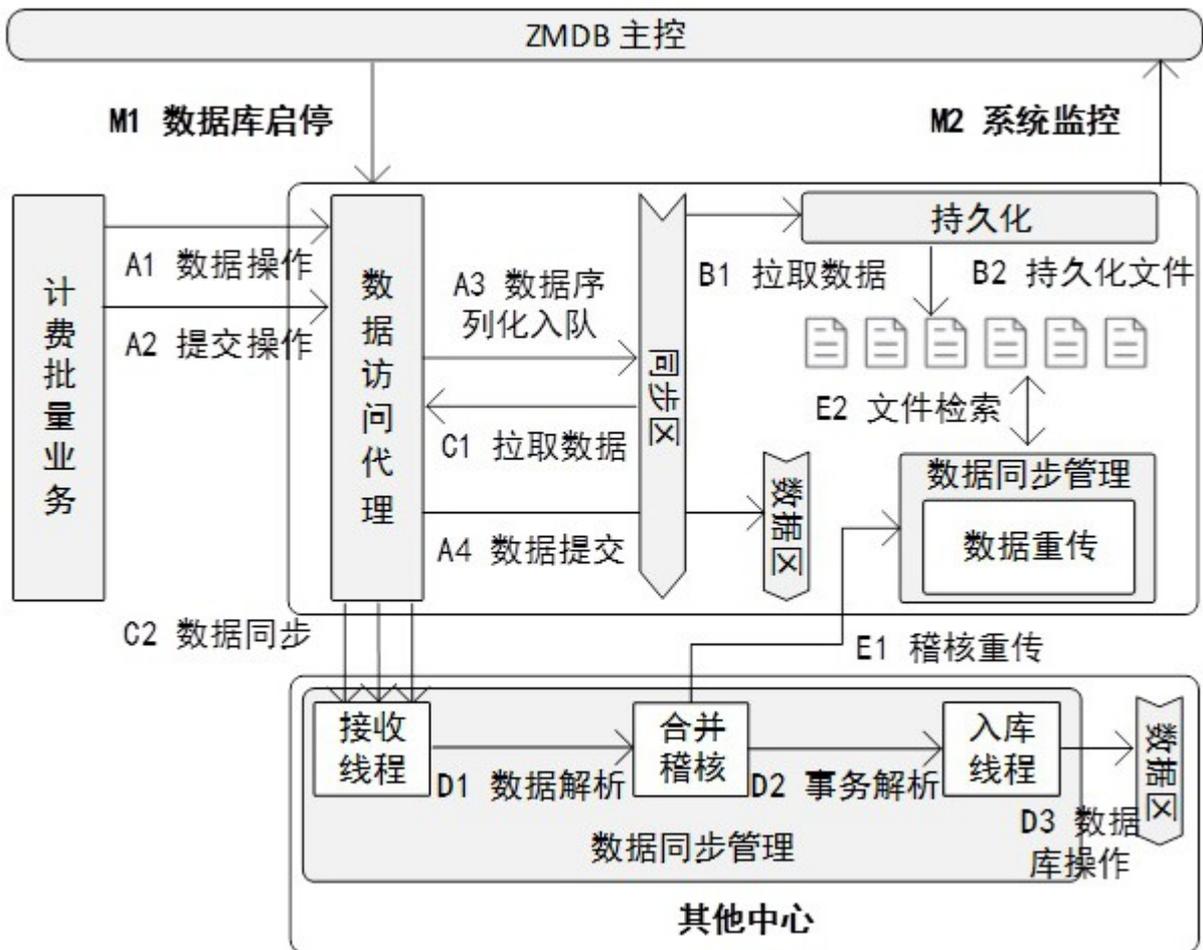


图 1

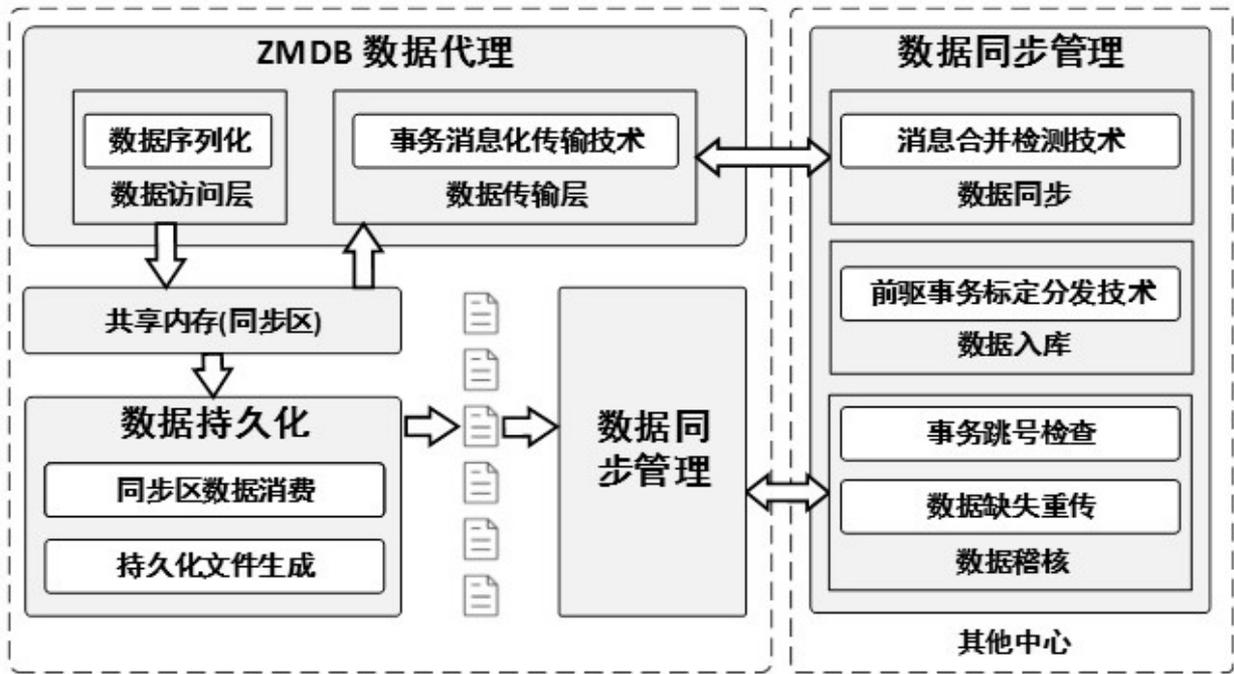


图 2

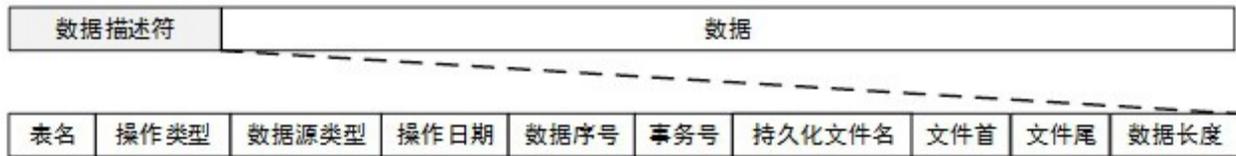


图 3



图 4

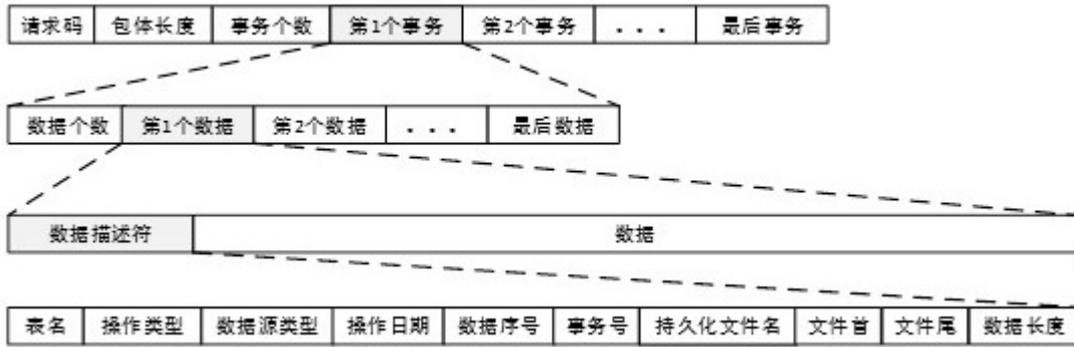


图 5

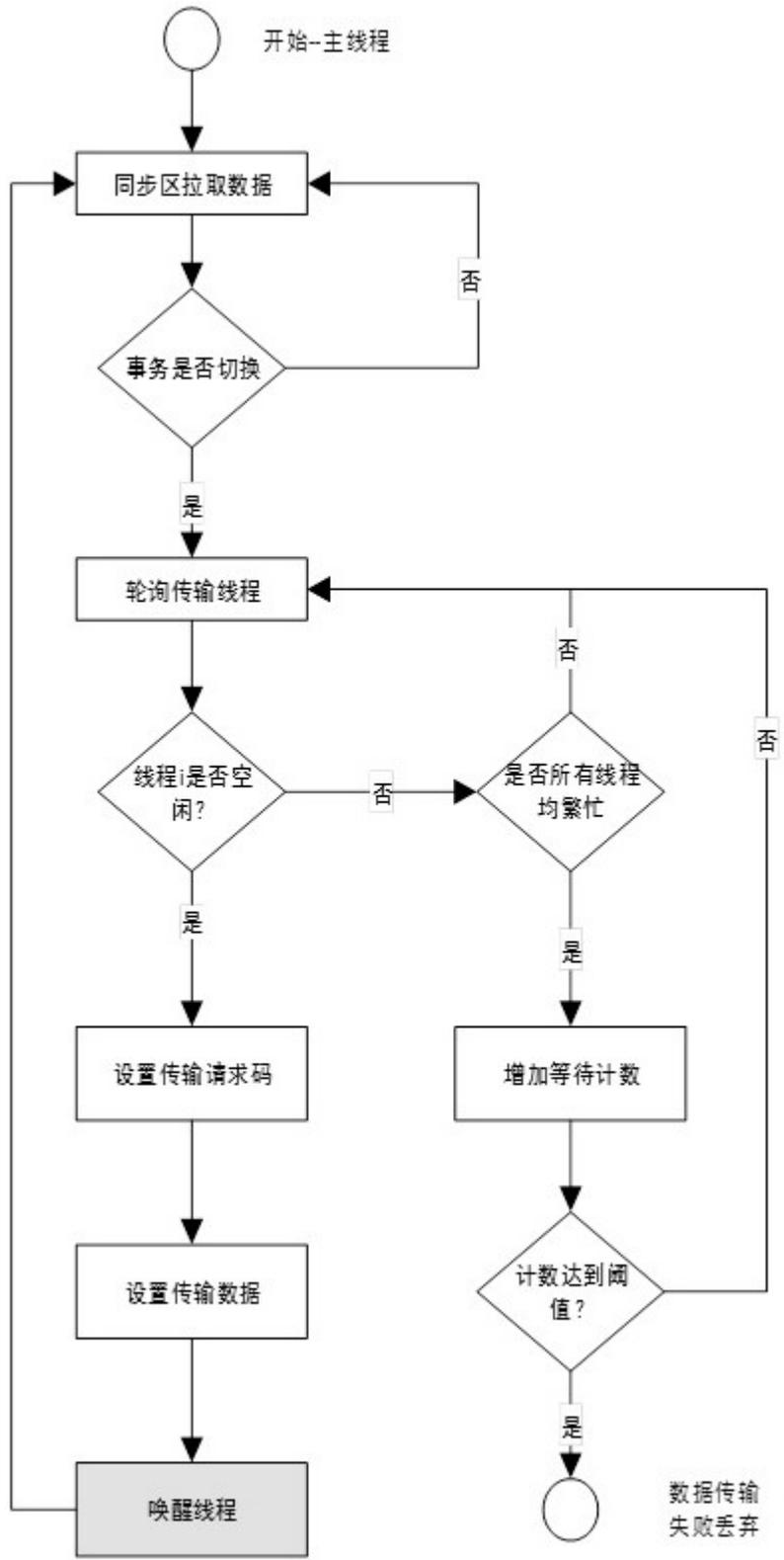


图 6(a)

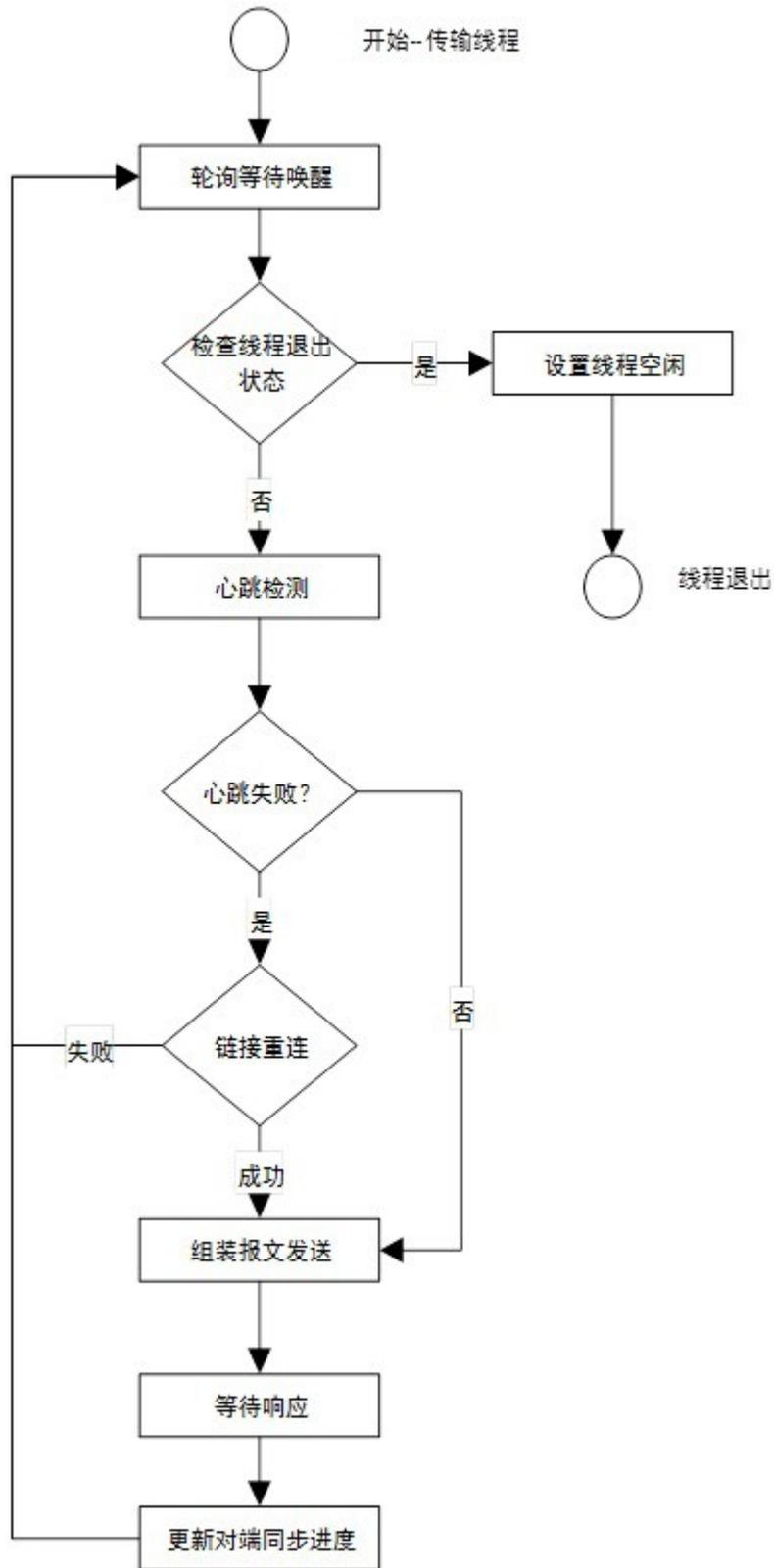


图 6 (b)

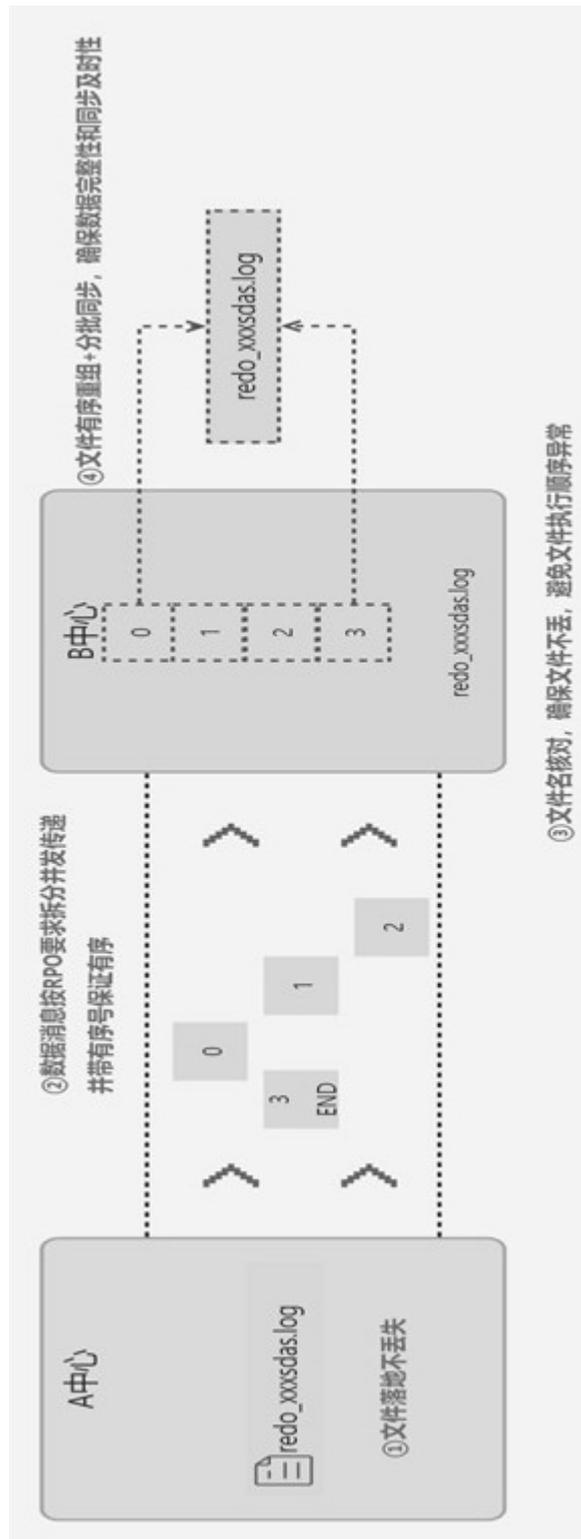


图 7

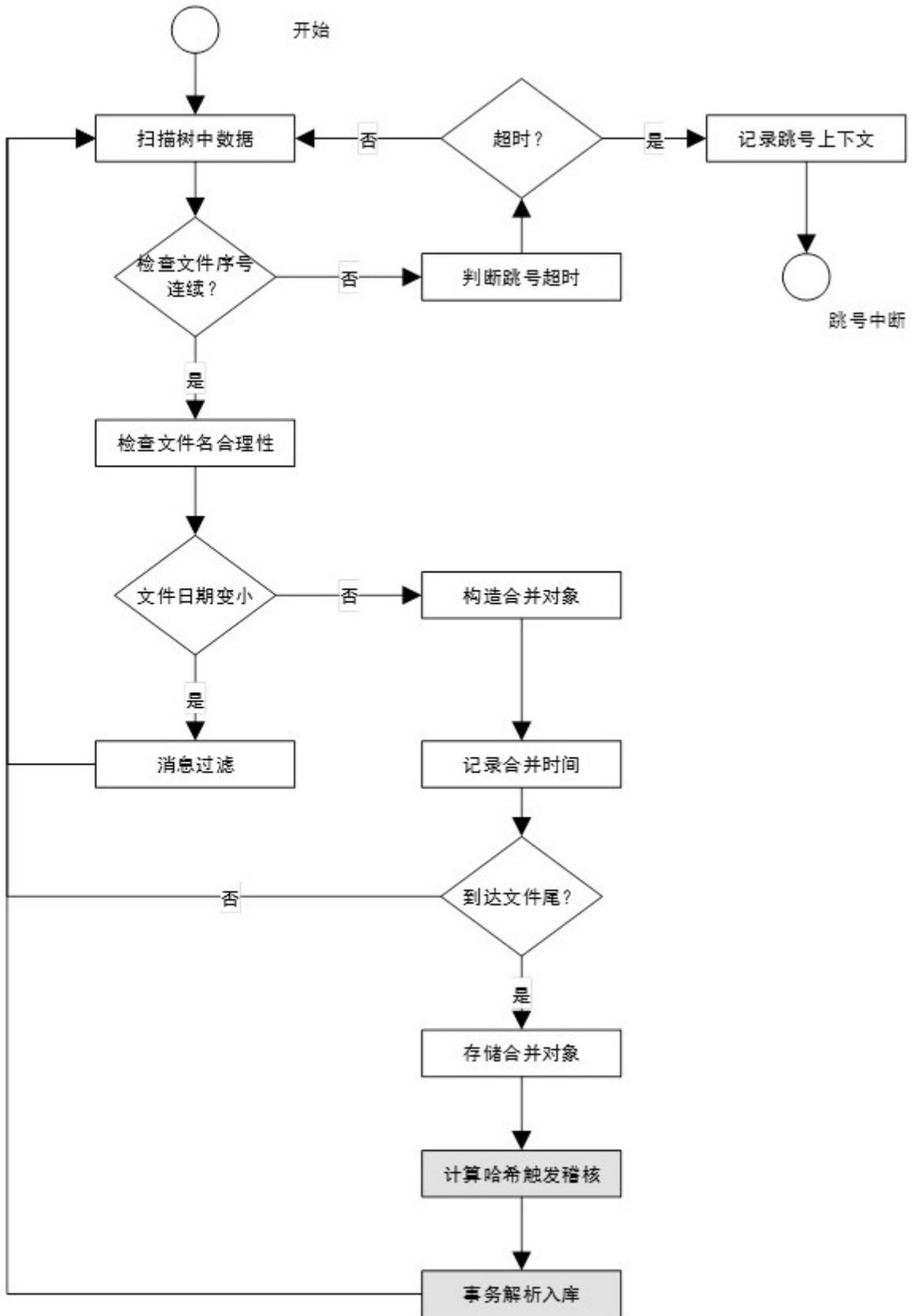


图 8

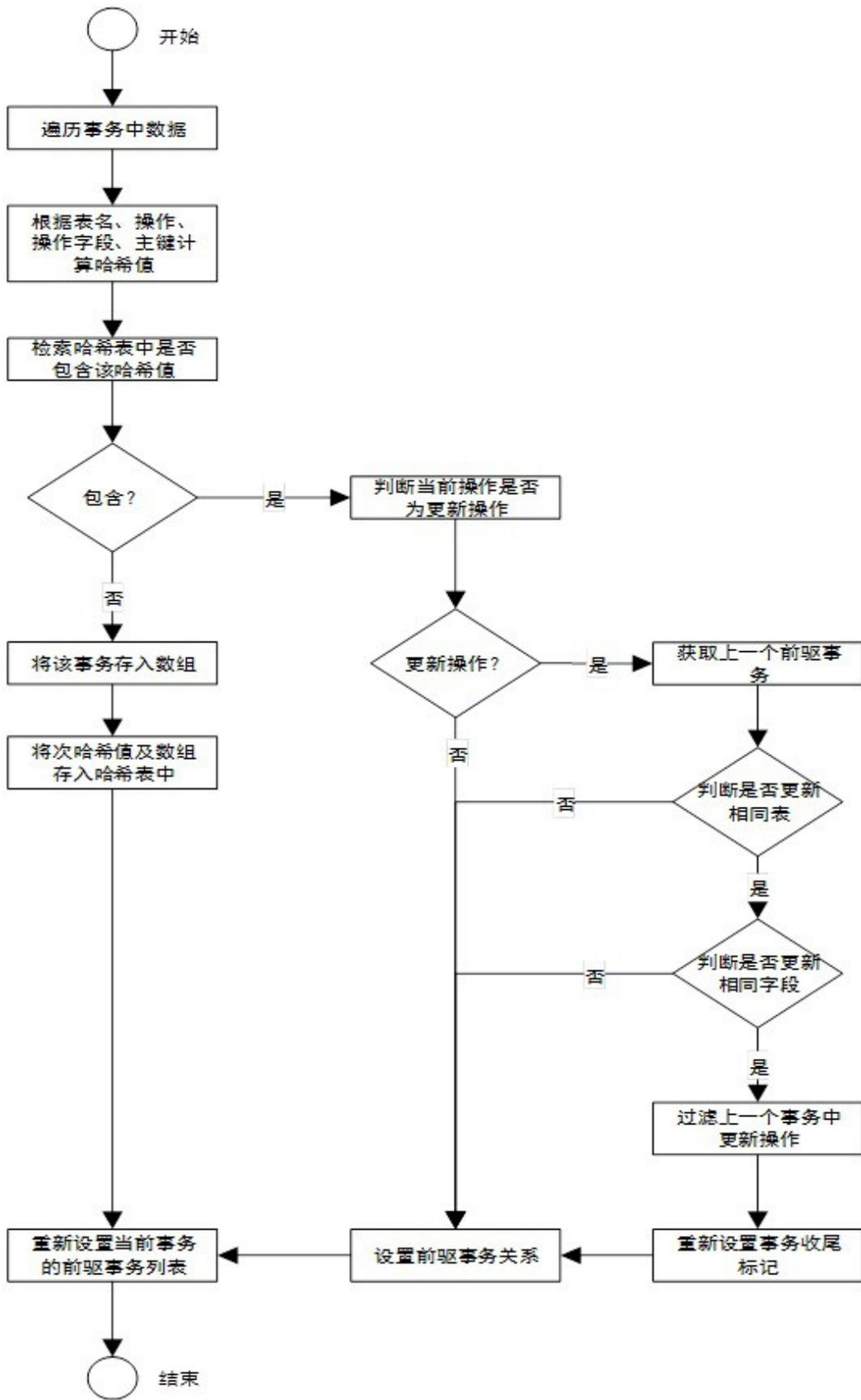


图 9