

# 基于区块链的多方 MDB 数据修复系统的设计与应用

陈彦翔<sup>1</sup>, 彭志强<sup>2</sup>

(1. 中国移动通信集团湖南有限公司, 湖南长沙 410001;  
2. 湖南工学院, 湖南衡阳 421200)

**摘要:**在分布式数据库环境中,确保数据一致性和不可篡改性始终是一大挑战。针对这一问题,提出了一种基于区块链技术的多方 MDB 内存库数据同步稽核方法。该方法依托区块链的去中心化和防篡改特性,将各个 MDB 内存库的数据通过区块链节点进行记录和同步,从而实现数据的高可靠性和可追溯性。在用户交互设计上,系统采用了 VUE3.0 框架和 IVIEW 组件库,为用户提供了现代化、直观的界面体验。同时,后台架构依赖 Nginx 集群和 Tomcat 服务器,确保了高并发场景下的数据传输效率和系统的稳定性。系统的整体架构包括数据上链、数据稽核和差异修复三大模块。首先,通过区块链节点对各个 MDB 内存库进行数据上链,保证了数据的不可篡改性和版本记录的完整性。其次,稽核模块通过智能合约自动比对各个区块链账本中的数据,确保各 MDB 内存库之间的数据同步性,并在检测到数据差异时触发预警机制。最后,差异修复模块通过区块链溯源功能,定位数据不一致的源头并自动修复差异,确保业务数据的准确性和一致性。实际应用表明,该方法有效提高了数据管理的效率和可靠性,显著减少了人工干预的需求,降低了系统运维成本。通过区块链技术的引入,系统进一步提升了业务数据管理的自动化和安全性,为复杂分布式系统中的数据管理提供了一种创新的解决方案。

**关键词:**分布式数据库;数据同步;区块链;数据稽核;差异修复

中图分类号:TP302.1

文献标识码:A

文章编号:1673-629X(2025)02-0207-06

doi:10.20165/j.cnki.ISSN1673-629X.2024.0322

## Design and Application of Multi-party MDB Data Repair System Based on Blockchain

CHEN Yan-xiang<sup>1</sup>, PENG Zhi-qiang<sup>2</sup>

(1. China Mobile Communications Group Hunan Co., Ltd., Changsha 410001, China;  
2. Hunan Institute of Technology, Hengyang 421200, China)

**Abstract:** Ensuring data consistency and immutability in distributed databases is a significant challenge. We propose a method for synchronizing and auditing data across multiple MDB (Memory Database) systems using blockchain technology to address this issue. According to the decentralization and tamper-proof characteristics of the blockchain, the proposed method records and synchronizes the data of each MDB memory library through the blockchain nodes, so as to achieve high reliability and traceability of the data. In the user interaction design, the system adopts VUE3.0 framework and IVIEW component library to provide users with modern and intuitive interface experience. At the same time, the backend architecture relies on Nginx cluster and Tomcat server to ensure data transmission efficiency and system stability in high concurrency scenarios. The system is structured around three key modules: data anchoring on the blockchain, data auditing, and discrepancy repair. First of all, the data is linked to each MDB memory library through the blockchain node, ensuring the immutability of the data and the integrity of the version record. Secondly, the auditing module automatically compares the data in the various blockchain ledgers through smart contracts, ensuring data synchronization between the various MDB memory libraries and triggering an early warning mechanism when data discrepancies are detected. Finally, the discrepancy repair module locates the source of data inconsistencies and automatically fixes the differences through the blockchain traceability function to ensure the accuracy and consistency of business data. The practical application shows that the proposed method can effectively improve the

收稿日期:2024-08-01

修回日期:2024-12-03

基金项目:湖南省教育基金项目(202401001561)

作者简介:陈彦翔(1989-),男,硕士,工程师,研究方向为应用系统开发、大数据、区块链应用;通讯作者:彭志强(1990-),男,硕士,讲师,研究方向为区块链应用、无线通信。

efficiency and reliability of data management, significantly reduce the need for manual intervention, and reduce the cost of system operation and maintenance. Through the introduction of blockchain technology, the system further improves the automation and security of business data management, providing an innovative solution for data management in complex distributed systems.

**Key words:** distributed database; data synchronization; blockchain; data audit; discrepancy repair

## 0 引言

中国移动作为全球领先的电信运营商之一,其业务运营离不开业务操作支撑系统(BOSS系统)的支持。BOSS系统是中国移动业务运营的核心,承担着管理和支持各项业务运营活动的重要职责<sup>[1]</sup>。随着通信技术的不断发展和用户数量的持续增长,BOSS系统的复杂性和重要性也日益突显。

BOSS系统在应对庞大而复杂的业务运营过程中面临着一系列挑战,其中最为突出的问题之一便是数据不一致性。由于中国移动的规模庞大,涉及的业务种类繁多,BOSS系统中存在着多个业务子系统,通常情况下,业务系统设置有执行不同业务逻辑的子系统,一些重要数据(如以用户版本作为资料、账单、资产)会分别存储于不同MDB(内存数据库)中,由各子系统独立使用。正常情况下,各子系统的MDB数据要保持一致。但各子系统执行的业务逻辑不同,经常要独立发起MDB数据的变更,这就需要各子系统的MDB内存库之间进行数据同步<sup>[2-3]</sup>。而数据同步并不是每次都能成功,一旦发生意外未能发现,会对业务运维带来极为严重的影响<sup>[4]</sup>。

针对中国移动BOSS系统中MDB数据不一致性的挑战,提出了一种创新的解决方案:基于区块链的MDB路由版本稽核与修复系统。该系统利用区块链技术和MDB内存库的特性,能够实现BOSS系统中数据一致性的实时监控和稽核,及时发现并修复数据版本差异,从而确保业务数据的准确性和一致性。

该文将深入探讨这一系统的设计原理、技术架构以及在中国移动BOSS系统中的应用场景。通过引入这一创新系统,能够为中国移动的业务运营提供更加可靠和高效的支持,推动中国移动在电信行业的领先地位。

## 1 区块链技术简介

区块链技术是一种去中心化的分布式账本技术,最初被设计用来支持比特币等加密货币的交易记录<sup>[5]</sup>。然而,随着时间的推移,人们意识到区块链技术具有更广泛的应用潜力,可以用于各种领域的数据管理和交易<sup>[6]</sup>。

在区块链技术中,数据被组织成一个个称为“区块”的数据块,每个区块包含了一定数量的交易记录以及前一个区块的哈希值<sup>[7]</sup>。这些区块通过密码学技

术链接在一起,形成了一个不可篡改的数据链,即“区块链”<sup>[8]</sup>。区块链的去中心化特性意味着它没有单一的控制中心,而是由网络中的多个节点共同维护和管理<sup>[9]</sup>。

区块链技术具有去中心化、不可篡改、透明、安全和智能合约等特性,它为数据管理和交易提供了一种全新的解决方案,正在逐渐改变着各行各业商业模式和运作方式<sup>[10]</sup>。

## 2 MDB概述和存在的问题

### 2.1 MDB与版本号概述

内存数据库(Memory Database, MDB)是一种将数据存储在内存中以提高访问速度和处理性能的数据库管理系统<sup>[11]</sup>。相比传统的磁盘存储数据库系统, MDB具有更高的读写速度和更低的延迟,适用于对数据实时性要求较高的应用场景<sup>[12]</sup>。BOSS系统中主要的MDB主要有4种:ROUTE\_MDB用于存放路由类数据;ABM\_MDB用来存放资产免费资源信息;USER\_MDB存放基础资料;APS\_MDB存放账单累积量等信息。MDB为话单批价、账务累账、实时计费、消费提醒、缴费查欠、信控管理、话费信息和资源查询等提供服务, MDB运行情况直接影响着业务缴费、计费话单处理和客服服务等关键电信业务<sup>[13]</sup>。

版本号记录了系统间业务数据的账户信息变更轨迹,为跨系统的分布式协调提供依据。版本号的存在确保了各业务数据归属账户变更轨迹计数器的梳理顺序,保证数据的准确性和实时性。版本号可以视为数据的时间戳,记录了数据的变更历史,对于追溯数据的变更轨迹和维护数据一致性至关重要。在分布式系统中,版本号能够帮助准确判断数据变更的先后顺序,避免数据冲突和不一致的情况发生。

### 2.2 MDB存在的问题

当前系统中MDB存在几大痛点:首先,由于各资料的维护是独立的,不同子系统之间缺乏有效的数据同步机制,导致数据差异无法及时发现。这种情况下,可能会出现数据不一致的问题,从而影响了业务流程和决策的准确性。其次,由于业务操作的复杂性,人工操作时往往容易遗漏或错误地触发业务变更,导致数据更新不及时或不完整,进而影响系统的正常运行<sup>[14]</sup>。此外,目前的系统缺乏对数据差异进行实时监测和预警的能力,一旦数据出现差异,往往需要等到问

题变得严重才能被发现,无法及时采取措施修复数据,可能会造成较大的损失。另外,当前系统也缺乏对数据变更进行溯源的能力,无法清晰地追踪数据变更的来源和过程,一旦出现数据不一致的情况,很难确定是由于何种原因引起的,也无法对问题进行有效的解决和防范<sup>[15]</sup>。最后,当前系统也缺乏对不同子系统之间数据进行统一管理和同步的能力,导致数据在不同子系统之间存在分散和不一致的情况,影响了系统的整体效率和运行效果。这些问题严重影响了系统的稳定性和数据的一致性,需要引入新的技术手段和解决方案来解决。

### 3 多方 MDB 版本稽核修复方法设计

#### 3.1 总体方案

该方案引入一个独立于业务系统的区块链,业务系统中的各个 MDB 内存库分别配置有区块链中的不同区块链节点,每个区块链节点分别配置有独立的区块链记账权,当 MDB 内存库对本地的 MDB 数据进行更新后,调用对应的区块链节点,将本地最新版本的 MDB 数据记录至区块链中;基于区块链中,每个区块对应一个版本的 MDB 数据,根据区块中的时间戳和版本号提供了 MDB 数据变更的溯源。在需要进行数据同步稽核时,可以调取各 MDB 内存库对应的区块链账本,通过区块链账本的比对即可判断出各 MDB 内存库是否数据同步。再将差异用户输出至修复模块进行修复。

该文提供一种多方 MDB 内存库的数据同步稽核方法,具体包括如下步骤:

(1)数据记录至区块链:每个 MDB 内存库在本地进行 MDB 数据更新后,调用本地对应的区块链节点,将更新后的 MDB 数据记录至区块链中。每个 MDB 内存库对应一个独立的区块链节点,具备独立的记账权,这确保了每个内存库可以通过其节点独立存储本地数据。每当数据更新时,系统会在对应的区块链节点中生成一个新的区块进行存证。

(2)数据同步校验:当触发预设条件时,每个 MDB 内存库调取本地和至少一个其他内存库的区块链账本,以校验数据同步情况。预先部署的智能合约用于同步稽核,每个内存库在数据迁移前调用智能合约,检验本地与其他内存库的数据是否同步。校验方法包括对比本地和其他内存库最新区块的默克尔树,若一致则判定数据同步,否则发出预警提示。预警提示的逻辑也被写入智能合约,当检测到数据不同步时,系统会发出预警。

(3)数据修复:将差异预警信息输出至修复模块进行数据修复。修复模块分析差异原因并执行修复操

作,确保数据的一致性和完整性。

该方案将业务系统的每个 MDB 内存库分别对应配置至独立的区块链节点,通过区块链记录了 MDB 内存库的每次变更,并且基于区块链的多副本分布式账本进行跨数据库的数据同步稽核,以此提高数据同步稽核的效率,节省了业务系统的运维成本,并提高了数据同步的准确性。

#### 3.2 数据上链

图 1 展示了区块的结构示意图。区块的结构包括区块头和区块体。区块头主要包含前一区块的哈希、版本号、时间戳和默克尔树的树根。区块体则包含默克尔树的叶子节点。

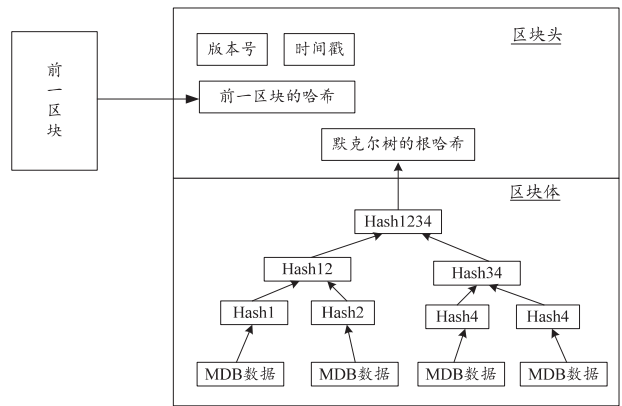


图 1 区块结构

在本方案中,区块的版本号对应于其记录的 MDB 数据的版本号,时间戳则记录了 MDB 数据的更新时间。也就是说,每个 MDB 内存库根据版本的变更,依次将一个区块记录在区块链网络中,并可追溯其更新时间。默克尔树最底层的叶节点包含存储的 MDB 数据或其哈希值,非叶子节点(包括中间节点和根节点)包含其子节点内容的哈希值。默克尔树的这种结构特性确保了底层数据的任何变化都会传递到父节点,最终影响树根的哈希值,这意味着根哈希值实际上代表了底层所有 MDB 数据的“数字摘要”。

本方案中,BOSS 系统的 MDB 内存库具体包括:路由内存库 ROUTE\_MDB、资料内存库 USER\_MDB、信控内存库 ABM\_MDB 以及账务内存库 APS\_MDB。这四个 MDB 内存库在正常情况下保持数据的一致性。

图 2 展示了系统场景中为 ROUTE\_MDB、USER\_MDB、ABM\_MDB 和 APS\_MDB 分别配置的区块链节点。每个 MDB 内存库的区块链节点拥有独立的记账权,通过区块链接口将每次更新后的 MDB 数据上链。每个版本的 MDB 数据以区块形式记录至区块链分布式共享账本。ROUTE\_MDB、USER\_MDB、ABM\_MDB 和 APS\_MDB 的原有数据传输和同步机制不变,依旧通过原有方式获取业务变更数据和业务资料数

据,并在修改版本号时,通过异步流程将更新后的数据上链。

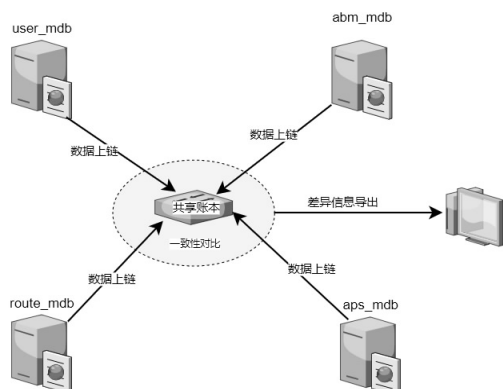


图 2 MDB 区块链结构

### 3.3 数据稽核

数据稽核过程主要包括以下几个步骤:

(1)触发稽核条件:每个MDB内存库在触发预设条件后(如定时任务或数据变更达到一定数量),会调取本地和至少一个其他MDB内存库对应的区块链节点的账本,以校验本地的MDB数据是否与其他MDB内存库的数据同步。

(2)调用智能合约:向本地区块链节点提交调用目标智能合约的交易,以执行数据同步稽核。智能合约会调取本地和其他MDB内存库的区块链账本,并比较最新区块的默克尔树。

(3)判断数据同步情况:通过比较各个区块链账本中最新区块的默克尔树,判断本地的MDB数据是否与其他MDB内存库的数据同步。如果默克尔树一致,则数据同步;否则,数据不同步,并发出预警提示。

以ROUTE\_MDB作为数据同步对象,具体流程如图3所示。

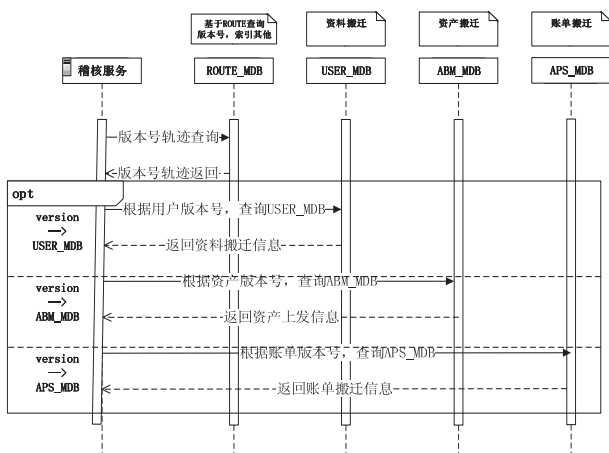


图 3 ROUTE\_MDB 同步示意图

(1)USER\_MDB在资料搬迁时,与ROUTE\_MDB版本比对。

(2)ABM\_MDB在资产上发时,与ROUTE\_MDB版本比对。

(3)APS\_MDB在账单搬迁时,与ROUTE\_MDB版本比对。

各MDB内存库进行版本比对的流程是:对对应的区块链节点提交携带本地MDB数据目标版本号的交易,由区块链节点调用目标智能合约,调取本地和其他MDB内存库对应的区块链账本。如果一致,则数据同步;若不一致,则生成差异预警信息,并根据区块链溯源能力,查找产生差异的节点信息。

### 3.4 数据修复

当收到差异预警信息时,系统会自动触发版本号修复流程,如图4所示。具体步骤如下:

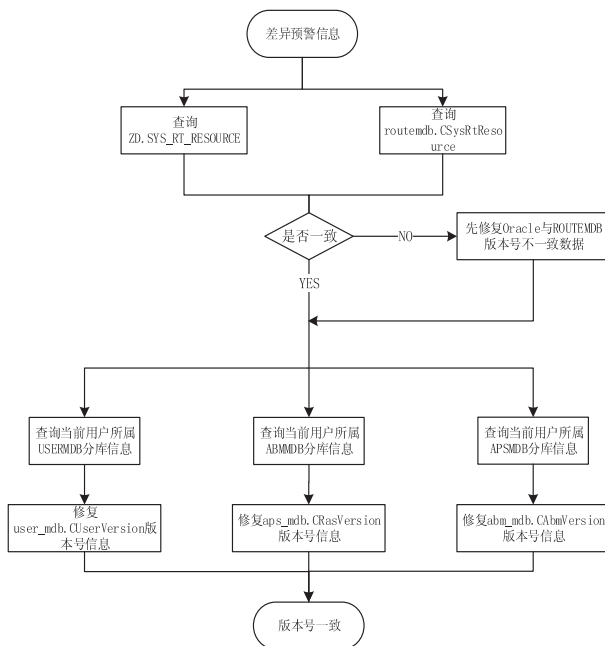


图 4 MDB 版本号修复示意图

(1)查询版本号:查询物理库ZD.SYS\_RT\_RESOURCE的版本号,并与ROUTE\_MDB中的route\_mdb.CSysRtResource版本号数据进行对比。

(2)判断和修复版本号:如果物理库ZD.SYS\_RT\_RESOURCE表中的版本号与route\_mdb.CSysRtResource版本号不一致,则将物理库中的版本号修复成route\_mdb.CSysRtResource的版本号。如果两者一致,则无需修复,直接进行下一步操作。

(3)获取差异信息:根据用户ID和差异预警信息,获取USER\_MDB、APS\_MDB、ABM\_MDB中存在差异的MDB对应分库信息。

(4)同步修复版本号:以route\_mdb.CSysRtResource的版本号为基准,将版本号同步修复至稽核预警的MDB中。

## 4 系统的应用

### 4.1 系统环境

该方案的硬件采用1台华为的计算型服务器

C12,其处理器为 Xeon Gold 5118 12Core ×2;内存 384G;磁盘空间为两块 480G SSD 和两块 960G 机械硬盘;网络为千兆网络。

软件方面,SERVER 端的主机操作系统为 BC-Linux for Euler;数据库使用 ORACLE 12c 和 AntDB;中间件使用 Tomcat、Kafka、Zookeeper 和 Eureka;编译器使用 Jdk1.8。CLIENT 客户端则使用 Win10 及以上系统,使用谷歌、火狐、Edge 等浏览器进行访问。

该方案建设底层区块链平台依托中移链 CMBaaS 区块链平台,新建 ROUTE\_MDB、USER\_MDB、ABM\_MDB 和 APS\_MDB 四个区块链节点,构建 MDB 版本号数据同步区块链网络。

### 4.2 系统架构

前端 UI 层是用户与系统交互的桥梁,采用 VUE3.0 框架、IVIEW 组件库、HTML/CSS,以及 Echarts 可视化库,提供直观、高效的交互体验。数据传输通过 JSON 格式进行序列化,确保标准性和可解析性。

交互层是前后端通信的枢纽,依托 Nginx 集群和 Tomcat 服务器,通过 POST/GET 请求和 Ajax 技术,实现高效、稳定的数据传输,确保快速响应用户请求和高效数据处理。

业务层是系统核心,包含数据上链、数据稽核、差异修复三大模块。数据上链模块负责实时采集和上链存储数据;数据稽核模块通过稽核条件和智能合约检查数据一致性;差异修复模块及时发现并修复数据差异。

业务数据层结合内存数据库和物理数据库,内存数据库如 USERDB、ABMDB 和 APSDB 用于高频访问的临时数据,物理数据库如 Oracle 和 AntDB 用于长期业务数据的持久化存储,确保数据的安全性和可靠性。

区块链层通过 API&SDK 接口、智能合约引擎、共识算法和共享账本,提供不可篡改、去中心化的分布式存储解决方案。P2P 网络技术确保区块链节点之间高

效、安全的通信。

基础设施层提供系统运行的硬件和网络环境支持,包括双中心资源池主机、BC-Linux for Euler 操作系统、网络管理和存储,确保系统高可用性、容灾能力、传输效率和存储安全性。

该系统设计充分利用区块链技术优势,结合现代化前端和高效稳定的后端交互机制,实现 MDB 路由版本的稽核与修复,保障数据一致性、安全性和可追溯性。系统的整体架构如图 5 所示。

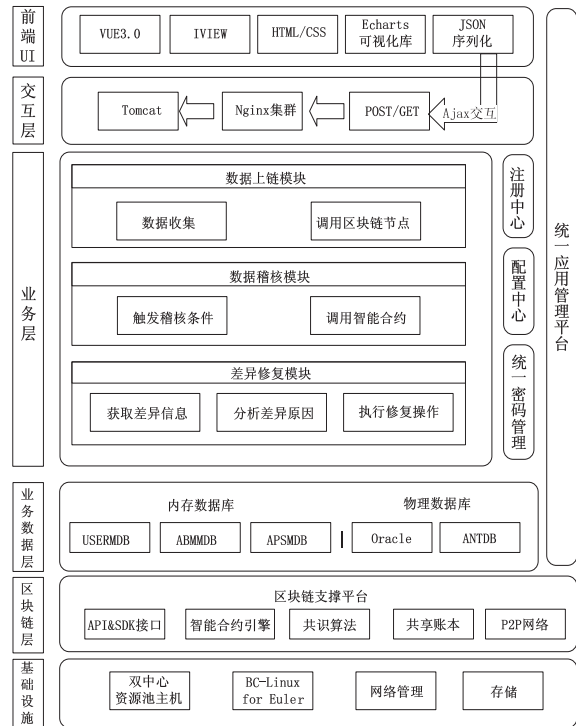


图 5 系统框架

### 4.3 运行结果与分析

在中移链 CMBaaS 区块链平台进行稽核后,如果发现数据差异,会返回本地预警信息,并通过 AntDB 数据库进行入库存储。如图 6 所示,USER\_ID 为 3105072511991031 的用户在 USER\_MDB 中存在差异,系统已将其入库记录。

```
select a.*,rowid from ti_f_audit_user_version_repair_05 a where resource_id='3105072511991031';
```

repair_id	resource_id	partition_id	repair_type	repair_code	diff_info	fail_message	done_tag	remark	deal_date
310702762507767808	3105072511991031	05	40001	20000	USERMDB版本号存在差异	9	[NULL]		2024-05-07 17:01:51.000

图 6 差异预警信息自动入库

根据平台数据,该用户的 11 分库存在 USER\_MDB 版本号差异,ROUTE\_MDB 的 A、B 分库的版本号均为 1,而该用户的 USER\_MDB 的 11 分库版本号为 0。

在系统稽核出具体的版本号差异后,按照修复流程对 USER\_MDB 版本进行更新。修复完成后,CLIENT 客户端的展示效果如图 7 所示。



图 7 版本强制修复轨迹入库展示

以 5 月为例,系统版本号稽核的总量为 984 546,其中差异总量为 1 594,本月差异比为 0.16%,其中差异较多的为 30003 结果状态码,其差异类型为 ROUTE\_MDB 与 USER\_MDB 不一致。系统稽核看板如图 8 所示。

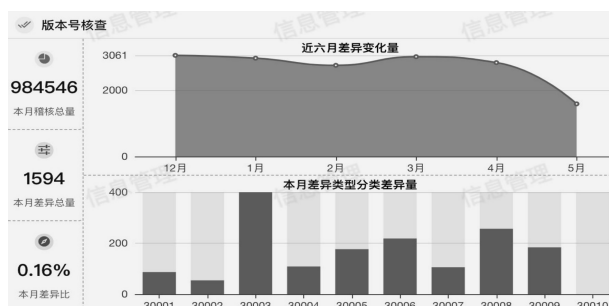


图 8 版本号稽核看板显示图

5 月版本号修复总量为 1 474,修复成功率为 92.5%。其中,差异较多的结果状态码为 40004 和 40006,分别定义为 USERMDB 与 APSMDB 版本号差异,以及 APSMDB 与 ABMMDB 版本号差异。从近 6 个月的运行情况看,版本号修复的数量趋势与稽核出的差异趋势基本重合,整体稽核修复水平较高,极大地减少了人工干预修复的工作量。版本号修复看板如图 9 所示。

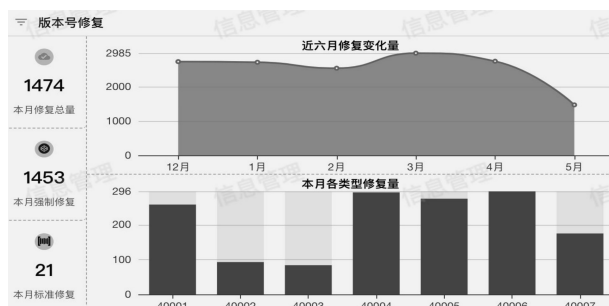


图 9 版本号修复看板显示图

## 5 结束语

该文提出的基于区块链技术的多方 MDB 内存库数据同步稽核方法,有效解决了分布式数据库环境下的数据一致性和不可篡改性问题。系统通过前端 VUE3.0 框架和 IVIEW 组件库提供现代化的交互界面,依托 Nginx 集群和 Tomcat 服务器实现高效稳定的数据传输。数据上链模块确保每次数据更新后的及时记录和存证;数据稽核模块利用智能合约对区块链账本中的数据进行比对,确保数据的一致性;差异修复模块在发现数据差异时自动触发修复流程,确保数据的准确性和完整性。实际应用结果表明,该方法显著提高了数据同步的准确性和效率,减少了人工干预,提高了数据管理的自动化水平和可靠性。未来的研究可以

进一步优化系统性能,提升智能合约的执行效率,并探索更多区块链技术在数据管理中的应用场景,以为分布式数据库的数据管理提供更加完善和高效的解决方案。

## 参考文献:

- [1] 刘宇宏. 新一代云原生 BOSS 系统的设计与实现[J]. 长江信息通信, 2023, 36(3): 148-151.
- [2] 储召乐, 罗永平, 金培权. 面向内存数据库的类字典树索引综述与性能比较[J]. 计算机学报, 2024, 47(9): 2009-2034.
- [3] 叶军, 文英, 蒙丽彬, 等. BOSS 内存数据库高可用性研究和应用[J]. 电信工程技术与标准化, 2016, 29(4): 12-17.
- [4] ZHANG Q, SONG H, ZHOU K, et al. A prefetching indexing scheme for in-memory database systems[J]. Future Generation Computer Systems, 2024, 156: 179-190.
- [5] 杨久华, 沈苏彬. 一种基于区块链的物联网访问控制方案[J]. 计算机技术与发展, 2023, 33(12): 136-142.
- [6] YANG P, REN J. A blockchain-based data auditing scheme with key-exposure resistance for IIoT[J]. Science China (Information Sciences), 2024, 67(2): 327-328.
- [7] EFTHYMIU M, MCCARTHY K, MARKOU C, et al. An exploratory research on blockchain in aviation: the case of maintenance, repair and overhaul (MRO) organizations[J]. Sustainability, 2022, 14(5): 1-17.
- [8] 于欣海, 黄欣哲, 梁海, 等. 基于区块链的高速联网收费数据传输应用研究[J]. 计算机技术与发展, 2023, 33(10): 51-58.
- [9] FAN K, LI F, YU H, et al. A blockchain-based flexible data auditing scheme for the cloud service[J]. Chinese Journal of Electronics, 2021, 30(6): 1159-1166.
- [10] 陈曦, 沈苏彬. 一种基于区块链的存储资源可信分配方法[J]. 计算机技术与发展, 2022, 32(2): 130-136.
- [11] 王坚, 田国良, 陈赛男. 分布式内存数据库大表 JOIN 性能提升研究[J]. 网络安全和信息化, 2024(3): 104-106.
- [12] YONGSIK K, SEONHO L, YEHYUN N, et al. DB[formula omitted]-tree: A new variant of B[formula omitted]-tree for main-memory database systems[J]. Information Systems, 2023, 119: 271-276.
- [13] 左康达, 孙知信. 基于区块链技术的土建工程项目管理平台[J]. 计算机技术与发展, 2021, 31(12): 187-192.
- [14] DIMOVSKI S A. Fault localization by abstract interpretation and its applications[J]. Journal of Computer Languages, 2024, 80: 1280-1288.
- [15] TAO Rui, ZHU Meng, CAO Haiyan, et al. Fine-grained cross-modal semantic consistency in natural conservation image data from a multi-task perspective[J]. Sensors, 2024, 24(10): 58-65.