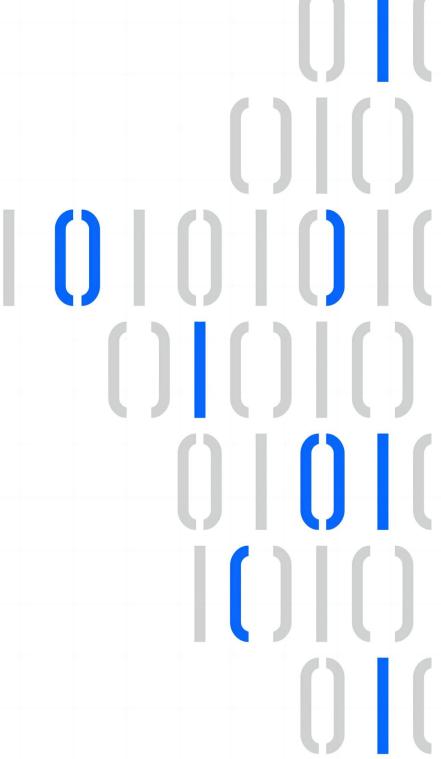
虚谷数据库 同步软件 V1.3.0 用户指南

文档版本 02

发布日期 2024-08-30





版权所有© 2024 成都虚谷伟业科技有限公司。

声明

未经本公司正式书面许可,任何企业和个人不得擅自摘抄、复制、使用本文档中的部分或全部内容,且不得以任何形式进行传播。否则,本公司将保留追究其法律责任的权利。

用户承诺在使用本文档时遵守所有适用的法律法规,并保证不以任何方式从事非法活动。不得利用本文档内容进行任何侵犯他人权益的行为。

商标声明



为成都虚谷伟业科技有限公司的注册商标。

本文档提及的其他商标或注册商标均非本公司所有。

注意事项

您购买的产品或服务应受本公司商业合同和条款的约束,本文档中描述的部分产品或服务可能不在您的购买或使用范围之内。由于产品版本升级或其他原因,本文档内容将不定期进行更新。

除非合同另有约定,本文档仅作为使用指导,所有内容均不构成任何声明或保证。

成都虚谷伟业科技有限公司

地址:四川省成都市锦江区锦盛路 138 号佳霖科创大厦 5 楼 3-14 号

邮编: 610023

网址: www.xugudb.com

前言

概述

本文档对同步软件的使用进行了详细的介绍。

读者对象

- 数据库管理员
- 数据库用户

符号约定

在本文中可能出现下列标志,它们所代表的含义如下。

| 符号 | 说明 |
|------------|-------------------------------------------------------|
| <u></u> 注意 | 用于传递设备或环境安全警示信息,若不避免,可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 |
| □ 说明 | 对正文中重点信息的补充说明。"说明"不是安全警示信息,不涉及人身、设备及环境伤害信息。 |

修改记录

| 文档版本 | 发布日期 | 修改说明 |
|------|------------|-------|
| 01 | 2024-03-15 | 第一次发布 |
| | | 接下页 |

| 文档版本 | 发布日期 | 修改说明 |
|------|------------|----------------------------|
| 02 | 2024-08-30 | cfg.properties 新增参数 dis- |
| | | able.binlog 和 isMax.binlog |

目录

| 1 | 产品 | 概述 | . 1 |
|---|-----|------------------------|-----|
| | 1.1 | 软件简介 | . 1 |
| | 1.2 | 产品构成 | . 1 |
| | 1.3 | 软件架构 | . 1 |
| | 1.4 | 核心指标 | . 2 |
| | 1.5 | 软件功能 | . 3 |
| | | 1.5.1 增量数据实时同步 | . 3 |
| | | 1.5.2 增量数据归档 | . 3 |
| | | 1.5.3 增量数据 kafka 集成 | . 3 |
| | | 1.5.4 DDL 和大对象同步 | . 3 |
| | | 1.5.5 事务支持 | . 4 |
| | | 1.5.6 元数据管理 | . 4 |
| | | 1.5.7 断点续传 | . 5 |
| | | 1.5.8 数据过滤 | . 5 |
| | | 1.5.9 运行日志 | . 5 |
| 2 | 配置 | 文件 | . 6 |
| | 2.1 | cfg.properties | . 6 |
| | 2.2 | dbcfg.properties | . 8 |
| | 2.3 | consumer.properties | . 9 |
| | 2.4 | producer.properties | 10 |
| | 2.5 | poll.properties | 11 |
| | 2.6 | application.properties | 12 |
| | 2.7 | config.json | 12 |
| 3 | 安装 | 部署 | 14 |
| | 3.1 | 运行环境 | 14 |
| | 3.2 | 部署模式 | 14 |
| | 3.3 | 安装与配置 | 14 |
| | | | |

| | | 3.3.1 | 源端配置 | 14 |
|---|-----|--------|---------------|------|
| | | 3.3.2 | XuguSyncer 配置 | 15 |
| | | 3.3.3 | 启动运行 | 22 |
| 4 | 同步 | 配置 | | 24 |
| | 4.1 | Binlog |] 同步至数据库 | 24 |
| | | 4.1.1 | 使用 Web 端配置启动 | . 24 |
| | | 4.1.2 | 使用 jar 配置启动 | 32 |
| | 4.2 | 同步至 | E Kafka | 38 |
| | | 4.2.1 | 使用 Web 端配置启动 | 38 |
| | | 4.2.2 | 使用 jar 包配置启动 | 41 |
| | 4.3 | Kafka | 同步至数据库 | 46 |
| | | 4.3.1 | 使用 Web 端配置启动 | 46 |
| | | 4.3.2 | 使用 jar 包配置启动 | 50 |
| | | 4.3.3 | 校验程序配置 | 55 |
| | | 4.3.4 | 主备配置 | 57 |
| 5 | 日常 | 运维 | | 58 |
| | 5.1 | binlog | ; 日志管理 | 58 |
| | | 5.1.1 | 开启 binlog 日志 | 58 |
| | | 5.1.2 | 关闭 binlog 日志 | 58 |
| | | 5.1.3 | 查询 binlog 日志 | 59 |
| | 5.2 | 订阅管 | 管理 | 61 |
| | | 5.2.1 | 创建订阅者 | 61 |
| | | 5.2.2 | 删除订阅者 | 62 |
| | 5.3 | 消费查 | 室询 | 62 |
| | | 5.3.1 | 系统表查询 | 62 |
| | | 5.3.2 | 同步软件日志查询 | 62 |
| | | 5.3.3 | Kafka 查询 | 63 |
| | | 5.3.4 | Kafka 数据验证 | 64 |
| | | 5.3.5 | Kafka 偏移推进 | 64 |
| | 5.4 | 常见问 | 可题 | 65 |

| 5.4.1 | 启动报错 | 65 |
|-------|------|----|
| 542 | 线程异常 | 65 |

产品概述

1.1 软件简介

虚谷数据库同步软件以虚谷分布式数据库作为数据源进行数据的全量和增量同步及分发,提供 界面化配置和启动。面对日益增长的数据量,虚谷数据库同步软件将虚谷分布式数据库中存储 的海量数据灵活的按需进行同步分发,满足用户基于海量数据的各类精细化数据服务需求,比 如专题库建立、重要数据归档备份、全量数据归档备份等。虚谷数据库同步软件增量同步基于 虚谷分布式数据库日志架构,完全不依赖于存储过程、函数、物化视图和触发器等,无需任何 代理,对源库零侵入性,完全不影响源库应用系统。

虚谷数据库同步软件作为虚谷数据库配套产品,不存在传统 ETL 的解决方案存在的技术壁垒、 低效率、高延迟等诸多难题。基于虚谷数据库架构,虚谷数据库同步软件具备高稳定性、高性 能、易使用的特点,核心功能包括:数据全量导出导入、异构库增量同步、增量变更查询、元 数据管理、DDL 同步、增量数据归档、KAFKA 集成和脚本化运行等。

1.2 产品构成

XuguSyncer 基于 Java 开发,产品包中主要包含软件主程序以及基础运行环境和基础配置文 件。其目录结构如下图1-1所示。

```
root
                         root root 4096 11月 20 11.
root root 1864362 11月 26 14:36 Encryption.jar
root root 4096 11月 26 11:02 lib
root root 4096 11月 26 11:03 log
root root 5191013 11月 26 14:36 pollbinlog.jar
root root 1573 11月 26 14:36 poll.properties
drwxr-xr-x 2 root root
           x--x 2 root root
                         root root
                                                       4096 11月
                                                                             26 14:36 redo
                                                      4096 11月 26 11:49 Statie
4096 11月 26 14:36 xgsyn.sh
1921 11月 26 14:36 XuguSyncer-1.3.0.jar
                      3 root root
                          root root
                          root root
```

图 1-1 目录结构

软件架构 1.3

虚谷数据库同步软件以虚谷数据库作为数据来源,实现全量数据和基于日志的增量数据获取、 解析、同步、分发功能。含基于 B/S 实现的 web 管理页面和支持服务器后台部署运行的控制台 版本、为用户提供易用、高效、稳定、可扩展的数据同步解决方案。

虚谷数据库同步软件架构如图1-2所示。

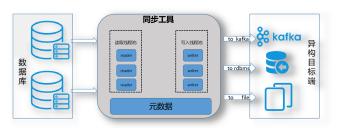


图 1-2 虚谷数据库同步软件架构图

1.4 核心指标

核心指标如表1-1所示。

表 1-1 虚谷数据库同步软件核心指标一览表

| 编号 | 指标项 |
|----|---------------------------|
| 1 | 基于数据库 binlog 零侵入性增量捕获 |
| 2 | 元数据自动管理 |
| 3 | 异构目标数据库支持 |
| 4 | 数据过滤 |
| 5 | 事务、DDL 和大对象支持 |
| 6 | 多线程、多机并行处理 |
| 7 | Kafka 大数据集成 |
| 8 | 日志捕获延迟小于 1s, 获取效率大于 10M/s |
| 9 | 关系数据库同步写入效率大于 10000 条/秒 |
| 10 | kafka 同步写入效率 20000 条/秒 |

1.5 软件功能

1.5.1 增量数据实时同步

依赖虚谷分布式数据库 binlog 日志文件进行实时数据同步,对源库零侵入性。采用多线程实时 检测捕获虚谷分布式数据库生产的事务操作 binlog 日志,将捕获的 binlog 日志结合对应表的元 数据信息反向生成标准 SQL 语句或以字段为单位的键值数据集,对应的多路数据同步写入器接 收 binlog 日志读取解析器生成的数据向目标数据库按事务序批量写入。

目前目标库支持主流关系型数据库,如 Oracle、MySQL 和虚谷数据库等。

1.5.2 增量数据归档

对数据库增量 binlog 日志进行归档存放实现自管理维护,从而不依赖于数据库 binlog 日志生命周期限制,实现数据库 binlog 日志的获取、压缩和存储。除支持将获取的 binlog 日志压缩后以二进制的方式存储外,同时支持将其解析翻译为标准 SQL 操作语句进行保存,SQL 语句归档文件可直接应用到其他关系型数据库。

1.5.3 增量数据 kafka 集成

虚谷数据库 binlog 在服务端以库为单位进行管理记载,即同一个逻辑库下所有对象的变更 binlog 日志项混合存储。为满足用户以表为单位进行 binlog 解析分发同步的需求,虚谷数据库 同步软件实时的将虚谷数据库记载的 binlog 以表为单位切分,然后发布到 kafka 上如1-3。用户可自由消费 kafka 上的虚谷数据库增量数据,因此虚谷数据库同步软件可为大数据集成业务提供数据服务。

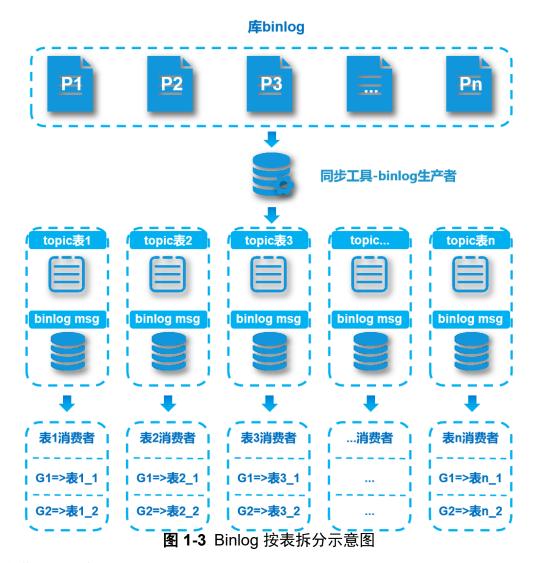
切分后 binlog 按表分发到 kafka 消息中间件,kafka 为每一个表建立 topic 以实现按表管理 binlog 日志。

□ 说明

topic 命名规则 [库名. 表名]。

1.5.4 DDL 和大对象同步

虚谷数据库的 binlog 日志采用原样记载 DDL 操作的 SQL 语句的方式,用户可从 binlog 日志中提取 DDL 操作的 SQL 语句。若目标端为关系型数据库,DDL 语句也可在目标库进行同步。由于 DDL 操作为重量级操作,所以默认情况下 DDL 同步不予开启,规避删除表、清理数据等破



坏性操作传递到目标库。

针对含大对象(clob、blob)数据的 binlog 日志行,仅支持小于 10 兆的大对象同步。

1.5.5 事务支持

软件从数据库获取已提交操作事务 binlog 日志,不会读取到未提交事务 (虚谷分布式数据库 binlog 日志为提交时写盘)。对读取到的所有事务 binlog 日志 (不论是小事务还是大事务) 进行全量解析,按 binlog 日志事务顺序执行保证同步后数据事务一致性。

1.5.6 元数据管理

软件解析数据库 binlog 日志时,可自动发现 DDL 操作 (如新建表、修改表结构、添加约束、添加索引等),对于新建表修改表结构操作 binlog 日志,将解析新表结构字段详细信息并加入元数据维护,对表元数据信息进行多版本管理,为 DML 操作 binlog 日志解析、反向生产标准 SQL 提供准确的元信息。

1.5.7 断点续传

软件异常导致数据丢失对数据同步来说是致命的,考虑同步软件本身和运行环境可能出现的故障 (断电、断网等),软件设计完善的断点续传机制,保证软件重启后同步数据的零丢失零重复。断点续传是一种在同步软件运行时保存点 (savepoint) 的机制,在写入目标库、kafka 后记载当前位置,并将其作为重启后的获取数据库 binlog 日志的起始位置,同时针对不同目标进行了专有设计。

Kafka

因 kafka 本身不具备数据重复检测,所以重启时工具从 kafka 拉取最后一条发送成功的消息,从消息中取出其在虚谷数据库 binlog 日志中对应的偏移量,同步程序结合保存点和该偏移量,忽略已发送到 kafka 的 binlog 项实现增量 binlog 日志数据不重复不丢失。

● 关系型数据库

目标端为关系型数据库时,断点续传采用同步软件自身保存点 (savepoint) 和关系型数据库唯一性约束结合的方式保证数据零丢失和零重复。

1.5.8 数据过滤

软件支持对数据的过滤,过滤条件包括表级过滤、列级过滤和行级过滤等。

1.5.9 运行日志

记载同步软件运行日志,包括实时同步进度、同步效率统计、错误日志等。同时支持以上日志 打包为 Json 格式数据发送到第三方运维系统,为用户统一运维监控提供服务。

2 配置文件

2.1 cfg.properties

全局运行时参数配置如表2-1所示,文件内分为五个部分:同步类型、源端数据库配置、Kafka 消息中间件配置、目标端文件配置和同步至数据库的配置。

表 2-1 用户手册

| 配置参数 | 说明 |
|-----------------------------|---------------------------------------------------------------------|
| #type in (db,kafka,file) | - |
| source.type=db | 源端类型 |
| aim.type=db | 目标端类型 |
| #binlog source config | 源端数据库配置信息 |
| source.log.db=db | 待同步库名 |
| source.db.usr=usr_sod | 源端用户名 |
| source.db.pwd=password | 源端用户名密码(通过 Encryption.jar 加密后的密码) |
| source.db.port=12345 | 源端端口号 |
| source.log.suber.name=sync2 | 同步使用的订阅名 |
| source.log.filter.T=null | 同步过滤表达式,过滤指定表。格式:模式 名:表名,多个表之间以&符号分割。例: s1:t1&s2:t2。(均使用英文字符) |
| source.log.filter.V=null | 同步过滤表达式, 过滤指定视图 |
| source.log.filter.P=null | 同步过滤表达式, 过滤指定存储过程 |
| source.log.filter.F=null | 同步过滤表达式, 过滤指定存储函数 |

| 配置参数 | 说明 |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------|
| source.log.forward=true | 与 source.log.filter 配合使用,若为 true,则同步 source.log.filter 中所配置的表。若为 false,则不同步 source.log.filter 中所配置的表。默认为 true |
| source.log.partion=8 | 虚谷数据库日志分区 |
| source.log.fetch.size=20 | 同步每次拉取日志大小 |
| source.log.ips=127.0.0.1 | 单事务最大行数 |
| source.commit_rows=1000 | 同步过滤表达式, 过滤指定存储函数 |
| ddl.support=false | 是否进行 DDL 同步 |
| redo.offset.server=true | 是否启用数据库端订阅者偏移量 |
| reconnect.num=-1 | 数据库重连次数,为-1 则为无限重连 |
| disable.binlog=false | true 目标端 binlog 不记录同步的数据,false 为记录 |
| isMax.binlog=false | false 从 binlog 原始位置开始读取,true 从最新的文件号的 0 偏移量开始读取 |
| # kafka source config | 源端 kafka 配置 |
| source.kafka.parath=8 | 每个 topic 的分区数 |
| source.topic=test.d2.tclob\\test.u2.t1 | │ │ 需要同步的 topic 名,以"\\"分隔 |
| case.sensitive=true | 用于同步至 kafka 时创建 topic 的大小写控制。若为 true,则创建的 topic 名大小写按照源端库实际大小写创建。若为 false,则创建的 topic 名全为小写 |
| #to kafka | 写入 Kafka 消息中间件配置 |

| 配置参数 | 说明 |
|-------------------------------|--------------------------------------------------------------------------------------------|
| zk.connect=192.168.2.225:2181 | zookeeper ip |
| topic.partition=8 | Kafka 每个 topic 的分区数 |
| repeat.pos.kafka=true | 是否依赖 Kafka 数据进行断点续传过滤重复 |
| case.sensitive=true | 用于同步至 kafka 时创建 topic 的大小写控制。若为 true,则创建的 topic 名大小写按照源端库实际大小写创建。若为 false,则创建的 topic 名全为小写 |
| #to db | 写入数据库 |
| writer.thd.num=8 | 同步时写入线程数(建议和 source.log.partion相同) |
| writer.db=xugu | 目标库类型(xugu,oracle,mysql) |
| writer.schema=SYSDBA | 指定目标端模式名(严格区分大小写) |
| writer.parathd.num=1 | 入库并发数 |
| # to Http | EI 信息 |
| url.ei= | 针对天境 EI,可不填 |
| task.id= | 针对天境 EI,可不填 |
| task.name= | 针对天境 EI,可不填 |

2.2 dbcfg.properties

数据库连接信息配置分为同步目标库和 binlog 日志库信息配置,如表2-2所示。

表 2-2 dbcfg.properties 详细配置

| 库信息 | 配置参数 | 说明 |
|---------------|----------------------------------------------------------------|------------------------------|
| 目标库连接信息 | target-db.user=SYSDBA | - |
| 目标库连接信息 | target-db.password=SYSDBA | 通过 Encryption.jar 加密后的 密码 |
| 目标库连接信息 | target-db.driverClass=com.xu gu.cloudjdbc.Driver | - |
| 目标库连接信息 | target- db.jdbcUrl=jdbc:xugu://1 9 2 .168.2.222:9930/dcc | - |
| 数据库日志记载节点连接信息 | binlog-db.user=SYSDBA | - |
| 数据库日志记载节点连接信息 | binlog-db.password=SYSDBA | 通过 Encryption.jar 加密后的 密码 |
| 数据库日志记载节点连接信息 | binlog- db.driverClass=com.xugu.clou djdbc.Driver | - |
| 数据库日志记载节点连接信息 | binlog- db.jdbcUrl=jdbc:xugu://1 9 2 .168.2.225:9930/dbb | - |

2.3 consumer.properties

Kafka 消费者参数配置如表2-3,具体参数说明参考 Kafka 文档。

表 2-3 消费者参数配置信息

| 文件 | 说明 |
|----------------------------------------------|----------------|
| bootstrap.servers=ip1:port1,ip2:port2,ip3:po | Kafka 连接信息按需配置 |
| rt3 | |
| group.id=getendpos | 建议保持默认 |
| auto.offset.reset=earliest | 建议保持默认 |
| enable.auto.commit=false | 建议保持默认 |
| auto.commit.interval.ms=1000 | 建议保持默认 |
| session.timeout.ms=30000 | 源端用户名 |
| max.partition.fetch.bytes=10485760 | 建议保持默认 |
| max.poll.records=10 | 建议保持默认 |
| max.poll.interval.ms=300000 | 建议保持默认 |
| key.deserializer=org.apache.kafka.common.s | 必须保持默认 |
| erialization.StringDeserializer | |
| value.deserializer=org.apache.kafka.commo | 必须保持默认 |
| n.serialization.ByteArrayDeserializer | |

2.4 producer.properties

Kafka 生产者参数配置如表2-4所示,具体参数说明参考 Kafka 文档。

表 2-4 生产者参数配置信息

| 文件 | 说明 |
|----------------------------------------------|----------------|
| bootstrap.servers=ip1:port1,ip2:port2,ip3:po | Kafka 连接信息按需配置 |
| rt3 | |

| 文件 | 说明 |
|------------------------------------------------|------------------|
| compression.type=none | 建议保持默认 |
| linger.ms=10 | 建议保持默认 |
| acks=all | 建议保持默认 |
| retries=8 | 建议保持默认 |
| batch.size=163840 | 建议保持默认 |
| buffer.memory=33554432 | 建议保持默认 |
| enable.idempotence=true | 建议保持默认 |
| key.serializer=org.apache.kafka.common.se | 必须保持默认 |
| rialization.StringSerializer | |
| value.serializer=org.apache.kafka.common.s | 必须保持默认 |
| erialization.ByteArraySerializer | |
| partitioner.class=com.sync.util.BinlogPartitio | 必须保持默认 |
| ner | |

2.5 poll.properties

pollbinlog 工具从 Kafka 提取数据使用的消费者配置文件, 如表2-5。

表 2-5 生产者参数配置信息

| 文件 | 说明 |
|-------------------------------------------------|----------------|
| bootstrap.servers=ip1:port1,ip2:port2,ip3:port3 | Kafka 连接信息按需配置 |
| group.id=getendpos | 建议保持默认 |
| auto.offset.reset=pollsql | 建议保持默认 |

| 文件 | 说明 |
|------------------------------------------------------------------------------------|--------|
| enable.auto.commit=false | 建议保持默认 |
| auto.commit.interval.ms=1000 | 建议保持默认 |
| session.timeout.ms=30000 | 建议保持默认 |
| max.partition.fetch.bytes=10485760 | 建议保持默认 |
| max.poll.records=10 | 建议保持默认 |
| max.poll.interval.ms=300000 | 建议保持默认 |
| key.deserializer=org.apache.kafka.common.s erialization.StringDeserializer | 必须保持默认 |
| value.deserializer=org.apache.kafka.commo n.serialization.ByteArrayDeserializer | 必须保持默认 |

2.6 application.properties

参数配置信息如表2-6所示。

表 2-6 参数配置信息

| 文件 | 说明 | |
|------------------|--------------------|--|
| server.port=9523 | 同步软件端口号,用户根据自身情况修改 | |

2.7 config.json

config.json 文件在 static 目录下,用户需要将其中的 IP 地址和端口号改为部属的服务器 IP, 以及和 application.properties 文件中一致的端口号,如图2-1所示。

```
"backend": {
    "rootUri": "http://192.168.2.218:9523"
}
```

图 2-1 config.json

3 安装部署

3.1 运行环境

表 3-1 同步软件运行环境最低配置表

| 编 号 | 项目 | 配置 |
|----------------|---------|---------------|
| 1 | 操作系统 | Windows、Linux |
| 2 | 内存 | 2G 以上 |
| 3 | 磁盘 | 1G 以上 |
| 4 | 网络 | 千兆、万兆网络 |
| 5 | Java 环境 | JDK1.8 及以上 |

3.2 部署模式

- 可部署于 Windows 和 Linux 平台,均提供 web 界面化的操作模式,通过界面配置同步文件、启停同步以及获取运行状态信息。
- 可直接手动修改配置文件(binlog 源端信息、目标写入端信息等),java -jar 启动同步工具 jar 包来开启同步。

3.3 安装与配置

3.3.1 源端配置

3.3.1.1 开启 binlog 日志

虚谷数据库 binlog 日志的开启与关闭需要数据库管理员 SYSDBA 来操作,因此,本例的 binlog 开启与关闭都是由 SYSDBA 登录到指定的用户库进行操作。使用控制台工具登录 SYSDBA 用户,执行命令如下:

● 开启:

```
exec dbms\_replication.create\_modify\_source('*','*');
```

● 查看:

```
select db\_name,reg\_modify from dba\_databases where db\_name
='DB\_TEST';
```

执行结果如图3-1所示。

```
SQL> exec dbms_replication.create_modify_source('*','*');

Execute successful.
Use time:315 ms.

SQL> select db_name,reg_modify from dba_databases where db_name='DB_TEST';

DB_NAME | REG_MODIFY |

Total 1 records.

Use time:5 ms.
```

图 3-1 开启 binlog 日志

3.3.1.2 创建订阅者

订阅者由系统管理员 SYSDBA 来管理,因此,订阅者的创建与删除都需要由 SYSDBA 来操作。本例由 SYSDBA 登录到 db_test 库,执行创建和删除订阅者。使用控制台工具登录 SYSDBA 用户,执行命令如下:

● 创建:

```
exec dbms\_replication.create\_subscriber('s\_test','u\_test')
;
```

● 查看:

```
select ds.subs\_name,du.user\_name from dba\_subscribers ds
    join dba\_users du using(user\_id) where du.user\_name='U\
    _TEST';
```

执行结果如图3-2所示。

3.3.2 XuguSyncer 配置

数据源端基础配置完成后,即可进行虚谷数据库同步软件基础配置,配置主要涉及 cfg.properties (全局运行时参数)和 dbcfg.properties (数据库连接信息)两个配置文件。

```
SQL> exec dbms_replication.create_subscriber('s_test','u_test');

Execute successful.
Use time:3 ms.

SQL> select ds.subs_name,du.user_name from dba_subscribers ds
2 join dba_users du using(user_id) where du.user_name='U_TEST';

SUBS_NAME | USER_NAME |

s_test| U_TEST|

Total 1 records.

Use time:2 ms.
```

图 3-2 创建订阅者

3.3.2.1 cfg.properties

全局运行时参数配置,文件内分为五个部分:同步类型、源端数据库配置、Kafka 消息中间件配置、目标端文件配置和同步至数据库的配置(需要根据实际进行配置)。

数据库密码加密(若使用 web 启动界面化配置的则不需要加密,直接在界面输入明文密码): 首先通过执行 Encryption.jar 对数据库密码进行加密,如图3-3所示。

第一个红框为数据库明文密码,第二个为加密后的密码。



参数配置如表3-2所示。

表 3-2 参数配置

| 配置参数 | 说明 |
|---------------------------|-----------|
| #type in (db,kafka,file) | - |
| source.type=db | 源端类型 |
| aim.type=db | 目标端类型 |
| #binlog source config | 源端数据库配置信息 |
| source.log.db=db | 待同步库名 |
| source.db.usr=usr_sod_sod | 源端用户名 |

| 配置参数 | 说明 |
|-----------------------------|---------------------------------------------------------------------------------------------------------------|
| source.db.pwd=password | 源端用户名密码(通过 Encryption.jar 加密后的密码) |
| source.db.port=12345 | 源端端口号 |
| source.log.suber.name=sync2 | 同步使用的订阅名 |
| source.log.filter.T=null | 同步过滤表达式,过滤指定表。格式:模式名:表名,多个表之间以&符号分割。例: s1:t1&s2:t2。(均使用英文字符) |
| source.log.filter.V=null | 同步过滤表达式, 过滤指定视图 |
| source.log.filter.P=null | 同步过滤表达式, 过滤指定存储过程 |
| source.log.filter.F=null | 同步过滤表达式, 过滤指定存储函数 |
| source.log.forward=true | 与 source.log.filter 配合使用,若为 true,则同步 source.log.filter 中所配置的表。若为 false,则不同步 source.log.filter 中所配置的表。默认为 true |
| source.log.partion=8 | 虚谷数据库日志分区 |
| source.log.fetch.size=20 | 同步每次拉取日志大小 |
| source.log.ips=127.0.0.1 | 单事务最大行数 |
| source.commit_rows=1000 | 同步过滤表达式, 过滤指定存储函数 |
| ddl.support=false | 是否进行 DDL 同步 |
| redo.offset.server=true | 是否启用数据库端订阅者偏移量 |
| reconnect.num=-1 | 数据库重连次数,为-1 则为无限重连 |
| disable.binlog=false | true 目标端 binlog 不记录同步的数据,false 为记录 |

| 配置参数 | 说明 |
|----------------------------------------|--------------------------------------------------------------------------------------------|
| isMax.binlog=false | false 从 binlog 原始位置开始读取,true 从最新的文件号的 0 偏移量开始读取 |
| # kafka source config | 源端 kafka 配置 |
| source.kafka.parath=8 | 每个 topic 的分区数 |
| source.topic=test.d2.tclob\\test.u2.t1 | 需要同步的 topic 名,以"\\"分隔 |
| case.sensitive=true | 用于同步至 kafka 时创建 topic 的大小写控制。若为 true,则创建的 topic 名大小写按照源端库实际大小写创建。若为 false,则创建的 topic 名全为小写 |
| #to kafka | 写入 Kafka 消息中间件配置 |
| zk.connect=192.168.2.225:2181 | zookeeper ip |
| topic.partition=8 | Kafka 每个 topic 的分区数 |
| repeat.pos.kafka=true | 是否依赖 Kafka 数据进行断点续传过滤重复 |
| case.sensitive=true | 用于同步至 kafka 时创建 topic 的大小写控制。若为 true,则创建的 topic 名大小写按照源端库实际大小写创建。若为 false,则创建的 topic 名全为小写 |
| #to db | 写入数据库 |
| writer.thd.num=8 | 同步时写入线程数(建议和 source.log.partion相同) |
| writer.db=xugu | 目标库类型(xugu,oracle,mysql) |
| writer.schema=SYSDBA | 指定目标端模式名(严格区分大小写) |
| writer.parathd.num=1 | 入库并发数 |

| 配置参数 | 说明 |
|------------|-------------|
| # to Http | EI 信息 |
| url.ei= | 针对天境 EI,可不填 |
| task.id= | 针对天境 EI,可不填 |
| task.name= | 针对天境 EI,可不填 |

3.3.2.2 dbcfg.properties

数据库连接信息配置,分为同步目标库和 binlog 日志库信息配置,根据实际情况配置源端和目标端用户信息即可,如表3-3所示。

表 3-3 虚谷 JDBC 实现接口类

| 库信息 | 配置参数 | 说明 |
|---------------|----------------------------------------------------------------|------------------------------|
| 目标库连接信息 | target-db.user=SYSDBA | - |
| 目标库连接信息 | target-db.password=SYSDBA | 通过 Encryption.jar 加密后的 密码 |
| 目标库连接信息 | target- db.driverClass=com.xugu.clou djdbc.Driver | - |
| 目标库连接信息 | target- db.jdbcUrl=jdbc:xugu://1 9 2 .168.2.222:9930/dcc | - |
| 数据库日志记载节点连接信息 | binlog-db.user=SYSDBA | - |
| 数据库日志记载节点连接信息 | binlog-db.password=SYSDBA | 通过 Encryption.jar 加密后的 密码 |

| 库信息 | 配置参数 | 说明 |
|---------------|----------------------------------------------------------------|----|
| 数据库日志记载节点连接信息 | binlog- db.driverClass=com.xugu.clou djdbc.Driver | - |
| 数据库日志记载节点连接信息 | binlog- db.jdbcUrl=jdbc:xugu://1 9 2 .168.2.225:9930/dbb | - |

3.3.2.3 Sync2column.json

数据库表列过滤配置文件。该配置文件为 json 格式,支持单表多列的指定,表列值仅支持单表单列单值的指定。若数据库 cata_case_sensitives 参数为 false,配置文件中模式名、表名、列名大小写均可;若为 true,则配置文件中模式名、表名、列名需要和系统表中一致。通过执行 sql :show cata_case_sensitive 查询。(仅支持 int、varchar、boolean、numric、smallint、datetime 类型的数据筛选。仅支持单列数据过滤筛选。)

配置文件格式:

- 指定表列入库
 - 时间数据 datetime 指定格式,仅支持秒后一位: "2021-02-22 02:00:00.0"
 - 时间类型 time 指定格式, 仅支持秒后一位: "02:10:10.0"
 - boolean 类型数据格式: " true" 或" false"

• 指定表某些列进行入库

```
{
    "schemaName1.table1":{
    "type": "table",
    "data": "a,b"}
} (指定表schemaName1.table1的a、b两列入库。模式名以源端表模式名为准)
```

山 说明

■ schemaName1.table1: 模式. 表名

■ type: 默认为 "table", 不可修改

■ data:需要入库的列名

- 指定表列的指定值符合时该记录入库
 - range(范围,针对时间、数字,闭区间)

示例 1: 字段 a 的值在 2020-10-10 00:00:00<=a<=2021-10-10 00:00:00 的行数据可入库

```
{
"SYSDBA.test\_a.a" : {
"data" : "2020-10-10 00:00:00.0,2021-10-10 00:00:00.0",
"type":"range"}
}
```

示例 2: 字段 b 的值在 1<=b<=4 或 6<=b<=9 的行数据可入库 (同一字段或条件用分号";"分隔)

```
{
"SYSDBA.test\_a.b" : {
"data" : "1,4;6,9",
"type":"range"}
}
```

■ fixed (多值匹配)

示例:字段 a 的值为"a"或"b"的行数据可入库

```
{
    "SYSDBA.test\_a.a" : {
    "data" : "a\&b",
    "type":"fixed"}
}
```

■ noequal (不包含匹配)

示例:字段 a 的值为"a"或"b"的行数据不入库

```
{
"SYSDBA.test\_a.a" : {
"data" : "a\&b",
"type":"noequal"}
}
```

■ vague (单个值模糊匹配)

示例:字段 a 的值包含"a"的行数据可入库

```
{
    "SYSDBA.test\_a.a" : {
    "data" : "a",
    "type":"vague"}
}
```

■ 单表多字段值匹配入库,字段 a 和 b 以";"分隔,数值和操作类型以"|&|"分隔示例:

```
{
    "SYSDBA.test\_a.a;b" : {
    "data" : "a|\&|1\&3",
    "type":"fixed|\&|range"}
    }//过滤test\_a表的a,b字段,a字段需固定值a,b字段需范围值1<=b
<=3
    ``e
```

3.3.3 启动运行

在启动同步程序前,需要做最后的确认。确认 binlog 日志文件是否已经存在,日志位于安装目录 XHOME/XMODI/DB00000000 下,若此目录下不存在以 DMF 为后缀的 binlog 日志文件,则不能启动同步程序,此时启动程序会报错。此时,我们只需登录到开启 binlog 日志记载的库,执行任意可记载的 SQL 语句即可。开启 binlog 日志记载后,若没有任何数据变更,系统不会生成相应的日志文件,所以需要先执行任意可记载的 SQL 语句,生成日志文件。

通用同步程序支持数据库之间的同步、导出到文件等同步方式,具体运行机制由 cfg.properties 中配置项 aim.type 决定。

同步软件启动有如下两种形式。

• 以 jar 包形式直接启动:

```
./xgsyn.sh start
```

以 Web 形式启动命令,提供 Web 页面修改启动同步软件:

```
./xgsyn.sh start\_web
```

端口号为配置文件 application.properties 中设置的端口号,用户可自定义设置

```
server.port=9523
```

启动成功后登录 Web 页面进行配置,Web 页面如图3-4所示。



图 3-4 Web 页面

启动完成后,可以通过日志文件查看程序运行状态。

● 查看是否启动同步软件命令:

```
./xgsyn.sh status
```

● 停止同步软件命令:

./xgsyn.sh stop

4 同步配置

4.1 Binlog 同步至数据库

4.1.1 使用 Web 端配置启动

以 Web 形式启动命令:

./xgsyn.sh start\ web

```
rwxr-xr-x 2 root root
                        4096 11月 29 14:29 config
rwxr-xr-x 2 root root
                        4096 11月
                                  29 14:29 dic
rwxr-xr-x 1 root root 1864362 11月
                                  29 14:29 Encryption.jar
                        4096 11月 29 14:29 lib
rwxr-xr-x 2 root root
                        4096 11月 29 14:29 log
rwxr-xr-x 2 root root
rwxr-xr-x 1 root root 5191013 11月 29 14:29 pollbinlog.jar
                        1573 11月 29 14:29 poll.properties
rwxr-xr-x 1 root root
                        4096 11月 29 14:29 proc
rwxr-xr-x 2 root root
                        4096 11月 29 14:29 redo
lrwxr-xr-x 2 root root
lrwxr-xr-x 3 root root
                        4096 11月 29 14:29 static
                        1921 11月 29 14:29 xgsyn.sh
rwxr-xr-x 1 root root
rwxr-xr-x 1 root root 236550 11月 29 14:29 XuguSyncer-1.3.0.jar
root@xugu224 xugu-sync-1.3.0-beat4]# ./xgsyn.sh start_web
root@xugu224 xugu-sync-1.3.0-beat4]# nohup: 把輸出追加到"nohup.out"
```

图 4-1 Web 启动截图

Binlog 同步至数据库,需要在源端类型和目标端类型均选择"库",并配置同步软件配置信息和数据源配置信息。

● 同步软件配置信息如图4-2所示(带 '*'的为必填项)。

表 4-1 同步软件配置信息

| 配置参数 | 说明 |
|----------|----------------------------------------------------------------------------------------------|
| 同步使用的订阅名 | 数据库的 binlog 订阅者名字,与配置文件保存一致 |
| 正向同步过滤 | 与表级同步过滤表达式配合使用,若选择 "启用",则同步表级同步过滤表达式中所 配置的表。若为"关闭",则不同步表级同 步过滤表达式中所配置的表。默认为"启 用" |

| 配置参数 | 说明 |
|----------------|----------------------------------------------------------------|
| 日志记载节点 IP | 源端数据变更收集节点(G 节点) |
| 同步每次拉取日志大小 | 同步每次从 binlog 拉取日志大小 |
| 写入线程数 | 入库线程数,需要与 Binlog 分区数对应一致 |
| 单事务最大行数 | 单事务最大行数 |
| 是否进行 DDL 同步 | 启用同步 DDL 和 DML 操作,关闭只同步 DML 操作 |
| 是否启用数据库端订阅者偏移量 | 选择启用,同步软件初次启动按照数据库 binlog 记载位置续传。选择关闭,则按照本 地 redo 记载位置续传 |
| 重连次数 | 数据库重连次数,为-1 则为无限重连 |
| EI 推送 | 针对天境 EI,可不填 |

● 数据源配置信息如图4-3所示。

表 4-2 数据源配置信息

| 库 | 配置参数 | 说明 |
|-----------|----------|----------------------------|
| #binlog 库 | 库名 | 源端库名 |
| #binlog 库 | 用户名 | 源端用户名 |
| #binlog 库 | 密码 | 源端用户名密码(web 端使用明文密码) |
| #binlog 库 | IP 地址 | 源端库 IP |
| #binlog 库 | 端口 | 源端数据库端口号 |
| #目标库 | 目标库数据库类型 | 目标库类型(xugu, oracle, mysql) |

| 库 | 配置参数 | 说明 |
|-------|------------|--------------------------------|
| # 目标库 | 目标库名 | 源端库名 |
| # 目标库 | 用户名 | 目标端用户名 |
| # 目标库 | 密码 | 目标端密码 |
| # 目标库 | IP 地址 | 目标端数据库 IP 地址 |
| # 目标库 | 端口 | 目标端数据库端口号 |
| #目标库 | 同步时写入线程数 | 同步时写入线程数,需要与 Binlog 分区数对应一致 |
| # 目标库 | 每个写入线程的并发数 | 入库并发数 |

● 数据过滤配置如图4-4所示。

表 4-3 数据过滤配置信息

| 配置参数 | 说明 |
|---------|----------------------------------------------------------|
| column | 列过滤,指定表的某些列同步。例:同步 SYSDBA.TEST1_1 表列 a 的数据 |
| range | 列值的范围过滤,针对时间和数字,闭区间。例:同步表 SYSDBA.ABC 列 a 的值为 1<=a<=3 的数据 |
| fixed | 多值匹配。例:同步表 SYSDBA.AA 列 AAA 等于"aa" 的数据 |
| noequal | 不包含匹配。例: 不同步表 SYS-DBA.ALL_TYPE 字段 A 的值为"b"的 行数据 |

| 配置参数 | 说明 |
|-------|---------------------------|
| vague | 单个值模糊匹配。例: 同步表 SYS- |
| | DBA.TEST1_2 字段 A 的值包含"c"的 |
| | 行数据 |

● 同步数据库对象信息配置如图4-5所示。

表 4-4 同步数据库对象信息

| 配置参数 | 说明 |
|------|------------------------------------------------------|
| 表 | 若同步工具配置页面的正向同步过滤选项为 是,则是选择需要同步的表,为否则是选择 不需要同步的 |
| 视图 | 若同步工具配置页面的正向同步过滤选项为 是,则是选择需要同步的,为否则是选择不 需要同步的 |
| 存储过程 | 若同步工具配置页面的正向同步过滤选项为 是,则是选择需要同步的,为否则是选择不 需要同步的 |
| 存储函数 | 若同步工具配置页面的正向同步过滤选项为 是,则是选择需要同步的,为否则是选择不 需要同步的 |

- 在搜索框输入模式名和表名(不区分大小写),然后点击搜索,列表展示结果筛选后的数据。
- 点击重置,清空搜索筛选条件,展示全部数据。
- 点击上传文件, 上传 xlsx 格式文件, 同时配置文件中的数据和上传的数据一致。
- 点击展示勾选数据,表格将展示配置文件中的数据。
- 点击展示全部数据,表格将展示库下所有的表对象数据。

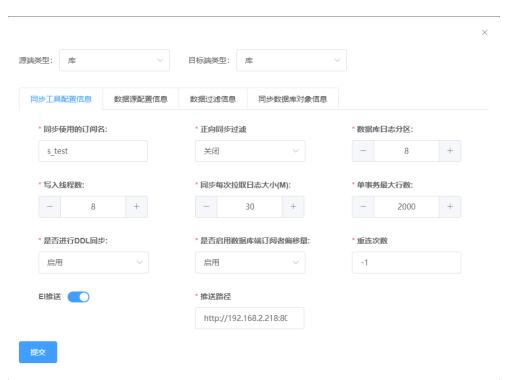


图 4-2 同步软件配置信息

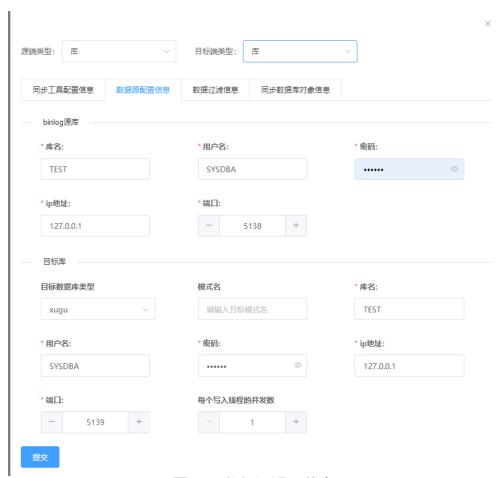


图 4-3 数据源配置信息



图 4-4 数据过滤

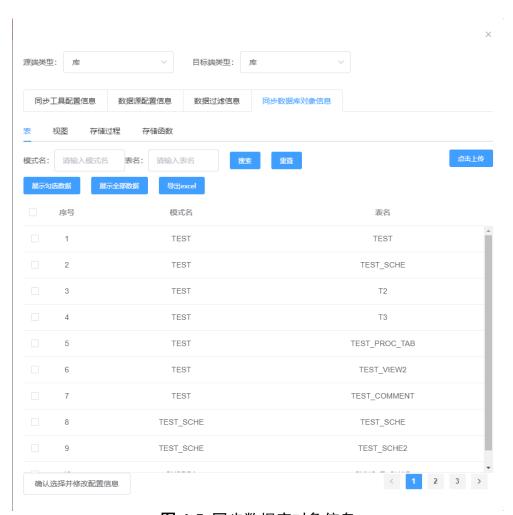


图 4-5 同步数据库对象信息

- 点击上传文件,上传 xlsx 格式文件。
- 点击导出 excel, 所有勾选的表对象数据会生成 xlsx 文件。
- 点击确认选择并修改配置信息,勾选的数据会同步到配置文件。

注:导入的 xlsx 文件格式分为两列,一列为模式名称,一列名表名称(需要区分大小写,与数据库模式和表信息一致)。如图4-6所示。

| 模式名称 | 表名称 |
|------------|----------------------------|
| SMS_SERVER | A112401 |
| SMS_SERVER | A1214 |
| SMS_SERVER | A121501 |
| SMS_SERVER | AAAAAA |
| SMS_SERVER | AZZH001 |
| SMS_SERVER | AZZH002 |
| SMS_SERVER | A_TEST01 |
| SMS_SERVER | A_TTY1 |
| SMS_SERVER | DATAINFO |
| SMS_SERVER | DIFFERENCE |
| SMS_SERVER | HIBERNATE_SEQUENCES |
| SMS_SERVER | JOBEXCEL |
| SMS_SERVER | JOB_INFO |
| SMS_SERVER | OBJECT_SCHEMA |
| SMS_SERVER | OBJECT_TABLE |
| SMS_SERVER | OBJECT_TABLEINFO |
| SMS_SERVER | PM_HIST_INSTANCE |
| SMS_SERVER | PM_HIST_JOB |
| SMS_SERVER | PM_HIST_PROCESS_TASK |
| SMS_SERVER | PM_HIST_VARIABLE |
| SMS_SERVER | PM_LOG |
| SMS_SERVER | PM_LOG_EMPOWER |
| SMS_SERVER | PM_LOG_TYPE |
| SMS_SERVER | PM_PROCESS_DEFINITION |
| SMS_SERVER | PM_PROCESS_DEFINITION_TYPE |

图 4-6 xlsx 文件格式

- 点击启动同步,同步开始,同时展示读线程和写线程的每个线程号的运行状态,如图4-7所示。
- 点击运行日志和错误日志查看同步日志信息,根据日志信息查看是否在正常同步,如图4-8所示。
- 点击停止同步, 停止当前同步任务, 如图4-9所示。
- 删除字典和删除 redo, 如图4-10所示。
 - 删除字典:字典为源端数据库元数据信息。若出现字典信息对应不上,可删除后重启同步。

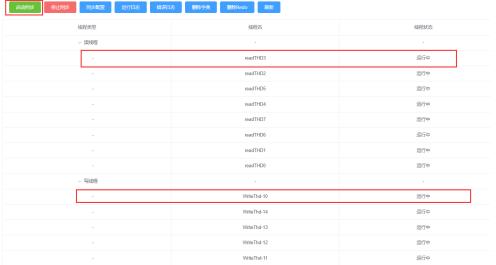


图 4-7 启动同步

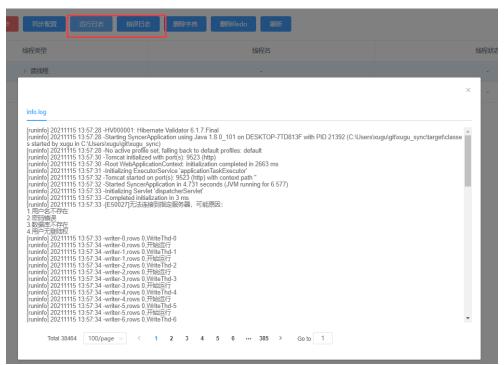


图 4-8 查看同步日志



图 4-9 停止同步

■ 删除 redo: redo 为同步软件本地所记载的读取位置信息。若位置出现偏差可删除后重启同步软件,则需要将同步软件配置信息中是否启用数据库端偏移量设为启用,否则同步软件会从 binlog 最开始的记载位置同步。



图 4-10 删除字典和删除 redo

4.1.2 使用 jar 配置启动

同步至数据库时,需要配置 cfg.properties 和 dbcfg.properties 配置文件。

● cfg.properties 具体配置 (同步类型、源端数据库配置和写入数据库) 如表4-5所示,可同步至 xugu、oracle、mysql。

表 4-5 cfg.properties 具体配置

| 配置参数 | 说明 |
|-----------------------------|-----------------------------------|
| #type in (db,kafka,file) | - |
| source.type=db | 源端类型 |
| aim.type=db | 目标端类型 |
| #binlog source config | 源端数据库配置信息 |
| source.log.db=db | 待同步库名 |
| source.db.usr=usr_sod | 源端用户名 |
| source.db.pwd=password | 源端用户名密码(通过 Encryption.jar 加密后的密码) |
| source.db.port=12345 | 源端端口号 |
| source.log.suber.name=sync2 | 同步使用的订阅名 |

| 配置参数 | 说明 |
|--------------------------|---------------------------------------------------------------------------------------------------------------|
| source.log.filter.T=null | 同步过滤表达式,过滤指定表。格式:模式 名:表名,多个表之间以&符号分割。例: s1:t1&s2:t2。(均使用英文字符) |
| source.log.filter.V=null | 同步过滤表达式, 过滤指定视图 |
| source.log.filter.P=null | 同步过滤表达式, 过滤指定存储过程 |
| source.log.filter.F=null | 同步过滤表达式, 过滤指定存储函数 |
| source.log.forward=true | 与 source.log.filter 配合使用,若为 true,则同步 source.log.filter 中所配置的表。若为 false,则不同步 source.log.filter 中所配置的表。默认为 true |
| source.log.partion=8 | 虚谷数据库日志分区 |
| source.log.fetch.size=20 | 同步每次拉取日志大小 |
| source.log.ips=127.0.0.1 | 单事务最大行数 |
| source.commit_rows=1000 | 同步过滤表达式, 过滤指定存储函数 |
| ddl.support=false | 是否进行 DDL 同步 |
| redo.offset.server=true | 是否启用数据库端订阅者偏移量 |
| reconnect.num=-1 | 数据库重连次数,为-1 则为无限重连 |
| disable.binlog=false | true 目标端 binlog 不记录同步的数据,false 为记录 |
| isMax.binlog=false | false 从 binlog 原始位置开始读取,true 从最新的文件号的 0 偏移量开始读取 |
| # kafka source config | 源端 kafka 配置 |
| source.kafka.parath=8 | 每个 topic 的分区数 |

| 配置参数 | 说明 |
|----------------------------------------|--------------------------------------------------------------------------------------------|
| source.topic=test.d2.tclob\\test.u2.t1 | 需要同步的 topic 名,以"\\"分隔 |
| case.sensitive=true | 用于同步至 kafka 时创建 topic 的大小写控制。若为 true,则创建的 topic 名大小写按照源端库实际大小写创建。若为 false,则创建的 topic 名全为小写 |
| #to kafka | 写入 Kafka 消息中间件配置 |
| zk.connect=192.168.2.225:2181 | zookeeper ip |
| topic.partition=8 | Kafka 每个 topic 的分区数 |
| repeat.pos.kafka=true | 是否依赖 Kafka 数据进行断点续传过滤重复 |
| case.sensitive=true | 用于同步至 kafka 时创建 topic 的大小写控制。若为 true,则创建的 topic 名大小写按照源端库实际大小写创建。若为 false,则创建的 topic 名全为小写 |
| #to db | 写入数据库 |
| writer.thd.num=8 | 同步时写入线程数(建议和 source.log.partion相同) |
| writer.db=xugu | 目标库类型(xugu,oracle,mysql) |
| writer.schema=SYSDBA | 指定目标端模式名(严格区分大小写) |
| writer.parathd.num=1 | 入库并发数 |
| # to Http | EI 信息 |
| url.ei= | 针对天境 EI,可不填 |
| task.id= | 针对天境 EI,可不填 |
| task.name= | 针对天境 EI,可不填 |

山 说明

- ■aim.type: 此参数为目标端类型,可选参数为 db、kafka、file, 此处配置为 db 即可。
- source.log.db 和 source.log.suber.name: 两个参数根据实际情况进行修改,分别为需要同步的库名和同步使用的订阅者名。
- source.log.ips: 为 binlog 日志记载节点的 IP, 支持多 IP 配置,可以根据实际情况修改 IP, 多个 IP 之间用逗号隔开。
- source.log.partion: 参数为源端虚谷数据库 binlog 日志分区数,此参数需要和数据库配置文件 xugu.ini 中的参数 modify_log_parti_num 保持一致,使用时根据实际配置修改此参数的值,数据配置默认为 8。另外,writer.thd.num 参数的值建议和source.log.partion 保持一致。
- source.log.filter: 根据实际同步需求进行修改, 其余参数使用默认值即可。

● dbcfg.properties 详细配置

■同步至虚谷数据库,如表4-6所示。

表 4-6 dbcfg.properties 详细配置(虚谷)

| 库信息 | 配置参数 | 说明 |
|---------------|----------------------------------------------------------------|------------------------------|
| 目标库连接信息 | target-db.user=SYSDBA | - |
| 目标库连接信息 | target-db.password=SYSDBA | 通过 Encryption.jar 加密后的 密码 |
| 目标库连接信息 | target- db.driverClass=com.xugu.clou djdbc.Driver | - |
| 目标库连接信息 | target- db.jdbcUrl=jdbc:xugu://1 9 2 .168.2.222:9930/dcc | - |
| 数据库日志记载节点连接信息 | binlog-db.user=SYSDBA | - |

| 库信息 | 配置参数 | 说明 |
|---------------|----------------------------------------------------------------|------------------------------|
| 数据库日志记载节点连接信息 | binlog-db.password=SYSDBA | 通过 Encryption.jar 加密后的 密码 |
| 数据库日志记载节点连接信息 | binlog- db.driverClass=com.xugu.clou djdbc.Driver | - |
| 数据库日志记载节点连接信息 | binlog- db.jdbcUrl=jdbc:xugu://1 9 2 .168.2.225:9930/dbb | - |

■同步至 MySQL 数据库,如表4-7所示。

表 4-7 dbcfg.properties 详细配置(MySQL)

| 库信息 | 配置参数 | 说明 |
|---------------|-------------------------------------------------------------------------------------------------|------------------------------|
| 目标库连接信息 | target-db.user=SYSDBA | - |
| 目标库连接信息 | target-db.password=SYSDBA | 通过 Encryption.jar 加密后的 密码 |
| 目标库连接信息 | target- db.driverClass=com.mysql.cj.j dbc.Driver | - |
| 目标库连接信息 | target- db.jdbcUrl=jdbc:mysql://192.1 68.2.85:3306/d2?useUnicode =true&characterEncoding=utf- 8 | - |
| 数据库日志记载节点连接信息 | binlog-db.user=SYSDBA | - |

| 库信息 | 配置参数 | 说明 |
|---------------|----------------------------------------------------------------|------------------------------|
| 数据库日志记载节点连接信息 | binlog-db.password=SYSDBA | 通过 Encryption.jar 加密后的 密码 |
| 数据库日志记载节点连接信息 | binlog- db.driverClass=com.xugu.clou djdbc.Driver | - |
| 数据库日志记载节点连接信息 | binlog- db.jdbcUrl=jdbc:xugu://1 9 2 .168.2.225:9930/dbb | - |

■同步至 Oracle 数据库,如表4-8所示。

表 4-8 dbcfg.properties 详细配置(MySQL)

| 库信息 | 配置参数 | 说明 |
|---------------|----------------------------------------------------------------------|------------------------------|
| 目标库连接信息 | target-db.user=u2 | - |
| 目标库连接信息 | target- db.password=MTlzNDU2 | 通过 Encryption.jar 加密后的 密码 |
| 目标库连接信息 | target-db.driverClass=oracle.j dbc.driver.OracleDriver | - |
| 目标库连接信息 | target- db.jdbcUrl=jdbc:oracle:thin:@//1 92.168.2.17:1521/orcl | - |
| 数据库日志记载节点连接信息 | binlog-db.user=SYSDBA | - |
| 数据库日志记载节点连接信息 | binlog-db.password=SYSDBA | 通过 Encryption.jar 加密后的 密码 |

| 库信息 | 配置参数 | 说明 |
|---------------|----------------------------------------------------------------|----|
| 数据库日志记载节点连接信息 | binlog- db.driverClass=com.xugu.clou djdbc.Driver | - |
| 数据库日志记载节点连接信息 | binlog- db.jdbcUrl=jdbc:xugu://1 9 2 .168.2.225:9930/dbb | - |

此配置文件为数据库连接信息配置,主要配置数据源和目标数据库基本连接信息,用户根据实际情况配置好即可。另外,此处源端的数据库用户必须是 cfg.properties 中订阅者绑定的用户。

cfg.properties 和 dbcfg.properties 配置修改完,检查无误后启动即可。

● 启动命令:

./xgsyn.sh start

● 查看是否启动同步软件命令:

./xgsyn.sh status

• 停止同步软件命令:

./xgsyn.sh stop

4.2 同步至 Kafka

4.2.1 使用 Web 端配置启动

以 Web 形式启动命令:

./xgsyn.sh start\ web

在 Web 端配置 binlog 同步至 kafka,需要在源端类型选择 "库",目标端类型选择 "kafka",并配置同步软件配置信息和数据源配置信息。同步软件配置信息同 Binlog 同步至数据库一致。

● 数据源配置信息如图4-11所示, 数据源配置信息如表4-9所示。



图 4-11 数据源配置信息

表 4-9 数据源配置信息

| 库 | 配置参数 | 说明 |
|-----------|----------------|-------------------------------------------------------------------------------------|
| #binlog 库 | 库名 | 源端库名 |
| #binlog 库 | 用户名 | 源端用户名 |
| #binlog 库 | 密码 | 源端用户名密码(web 端使用明文密码) |
| #binlog 库 | IP 地址 | 源端库 IP |
| #binlog 库 | 端口 | 源端数据库端口号 |
| #kafka | zkConnect | kafka 的 zookeeper 地址加端 口号 |
| #kafka | case Sensitive | 用于同步至 kafka 是创建 topic 的大小写控制。若为启用,则创建的 topic 名大小写按照源端库实际大小写创建。若为关闭,则创建的 topic 名全为小写 |
| #kafka | topicPartition | kafka 分区数 |
| #kafka | repeatPosKafka | 是否依赖 Kafka 数据进行断点 续传过滤重复 |
| #kafka | IP 地址 | 目标端数据库 IP 地址 |

● 消费者配置信息如图4-12所示, consumer.properties 消费者配置信息如表4-10所示。

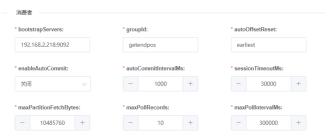


图 4-12 消费者配置信息

表 4-10 consumer.properties 消费者配置

| 配置参数 | 说明 |
|------------------------------------------------|----------------|
| bootstrapServers=ip1:port1,ip2:port2,ip3:port3 | Kafka 连接信息按需配置 |
| groupId=getendpos | 建议保持默认 |
| autoOffsetReset=earliest | 建议保持默认 |
| enableAutoCommit=false | 建议保持默认 |
| autoCommitIntervalMs=1000 | 建议保持默认 |
| sessionTimeoutMs=30000 | 建议保持默认 |
| maxPartitionFetchBytes=10485760 | 建议保持默认 |
| maxPollRecords=10 | 建议保持默认 |
| maxPollIntervalMs=300000 | 建议保持默认 |

● 生产者配置信息如如图4-13所示, producer.properties 生产者配置信息如表4-11所示。

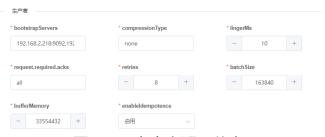


图 4-13 生产者配置信息

表 4-11 producer.properties 生产者配置

| 配置参数 | 说明 |
|------------------------------------------------|----------------|
| bootstrapServers=ip1:port1,ip2:port2,ip3:port3 | Kafka 连接信息按需配置 |
| compressionType=none | 建议保持默认 |
| lingerMs=10 | 建议保持默认 |
| acks=all | 建议保持默认 |
| retries=8 | 建议保持默认 |
| batchSize=163840 | 建议保持默认 |
| bufferMemory=33554432 | 建议保持默认 |
| enableIdempotence=true | 建议保持默认 |

4.2.2 使用 jar 包配置启动

使用 jar 形式直接启动同步软件同步至 Kafka 时,需配置 cfg.properties、consumer.properties和 producer.properties。

● cfg.properties 具体配置如表4-12所示。

表 4-12 cfg.properties 具体配置

| 配置参数 | 说明 |
|--------------------------|-----------|
| #type in (db,kafka,file) | - |
| source.type=db | 源端类型 |
| aim.type=db | 目标端类型 |
| #binlog source config | 源端数据库配置信息 |
| source.log.db=db | 待同步库名 |
| source.db.usr=usr_sod | 源端用户名 |

| 配置参数 | 说明 |
|-----------------------------|---------------------------------------------------------------------------------------------------------------|
| source.db.pwd=password | 源端用户名密码(通过 Encryption.jar 加密后的密码) |
| source.db.port=12345 | 源端端口号 |
| source.log.suber.name=sync2 | 同步使用的订阅名 |
| source.log.filter.T=null | 同步过滤表达式,过滤指定表。格式:模式名:表名,多个表之间以&符号分割。例:s1:t1&s2:t2。(均使用英文字符) |
| source.log.filter.V=null | 同步过滤表达式, 过滤指定视图 |
| source.log.filter.P=null | 同步过滤表达式, 过滤指定存储过程 |
| source.log.filter.F=null | 同步过滤表达式, 过滤指定存储函数 |
| source.log.forward=true | 与 source.log.filter 配合使用,若为 true,则同步 source.log.filter 中所配置的表。若为 false,则不同步 source.log.filter 中所配置的表。默认为 true |
| source.log.partion=8 | 虚谷数据库日志分区 |
| source.log.fetch.size=20 | 同步每次拉取日志大小 |
| source.log.ips=127.0.0.1 | 单事务最大行数 |
| source.commit_rows=1000 | 同步过滤表达式, 过滤指定存储函数 |
| ddl.support=false | 是否进行 DDL 同步 |
| redo.offset.server=true | 是否启用数据库端订阅者偏移量 |
| reconnect.num=-1 | 数据库重连次数,为-1 则为无限重连 |
| disable.binlog=false | true 目标端 binlog 不记录同步的数据,false 为记录 |

| 配置参数 | 说明 |
|-----------------------------------------|--------------------------------------------------------------------------------------------|
| isMax.binlog=false | false 从 binlog 原始位置开始读取,true 从最新的文件号的 0 偏移量开始读取 |
| # kafka source config | 源端 kafka 配置 |
| source.kafka.parath=8 | 每个 topic 的分区数 |
| source.topic=test.d2.tclob \\test.u2.t1 | 需要同步的 topic 名,以"\\"分隔 |
| case.sensitive=true | 用于同步至 kafka 时创建 topic 的大小写控制。若为 true,则创建的 topic 名大小写按照源端库实际大小写创建。若为 false,则创建的 topic 名全为小写 |
| #to kafka | 写入 Kafka 消息中间件配置 |
| zk.connect=192.168.2.225:2181 | zookeeper ip |
| repeat.pos.kafka=true | 是否依赖 Kafka 数据进行断点续传过滤重复 |
| case.sensitive=true | 用于同步至 kafka 时创建 topic 的大小写控制。若为 true,则创建的 topic 名大小写按照源端库实际大小写创建。若为 false,则创建的 topic 名全为小写 |
| #to db | 写入数据库 |
| writer.thd.num=8 | 同步时写入线程数(建议和 source.log.partion相同) |
| writer.db=xugu | 目标库类型(xugu, oracle, mysql) |
| writer.schema=SYSDBA | 指定目标端模式名(严格区分大小写) |
| writer.parathd.num=1 | 入库并发数 |
| # to Http | EI 信息 |

| 配置参数 | 说明 |
|------------|-------------|
| url.ei= | 针对天境 EI,可不填 |
| task.id= | 针对天境 EI,可不填 |
| task.name= | 针对天境 EI,可不填 |

□ 说明

同步至 Kafka 需要修改 cfg.properties 中的源端数据库信息和 Kafka 中间件信息。配置 aim.type 参数为 Kafka。源端数据库信息 (# binlog source config) 配置和源端配置一致,只需按照实际情况修改库名、订阅者、过滤表达式和日志记载节点 IP 即可。然后,Kafka 根据生产环境的配置,将 zk.connect 修改为 zookeeper 的 IP 端口,数据库连接使用订阅者 s 2kafka 绑定的用户 u 2kafka,指定任意一个节点的 IP 即可。

• producer.properties 生产者配置如表4-13所示。

表 4-13 producer.properties 生产者配置

| 配置参数 | 说明 |
|------------------------------------------------------------------------|----------------|
| bootstrapServers=ip1:port1,ip2:port2,ip3:port3 | Kafka 连接信息按需配置 |
| compressionType=none | 建议保持默认 |
| lingerMs=10 | 建议保持默认 |
| acks=all | 建议保持默认 |
| retries=8 | 建议保持默认 |
| batchSize=163840 | 建议保持默认 |
| bufferMemory=33554432 | 建议保持默认 |
| enableIdempotence=true | 建议保持默认 |
| key.serializer=org.apache.kafka.common.s e rialization.StringSerialize | 必须保持默认 |

| 配置参数 | 说明 |
|--------------------------------------------------------------------------------|--------|
| value.serializer=org.apache.kafka.common.s erialization.ByteArraySerializer | 必须保持默认 |
| partitioner.class=com.sync.util.BinlogPartitio | 必须保持默认 |

山 说明

按照实际部署情况,将 bootstrap.servers 修改为 Kafka 集群的 IP 和端口,其余参数保持默认。

● consumer.properties 消费者配置如表4-14所示。

表 4-14 consumer.properties 消费者配置置

| 配置参数 | 说明 |
|-------------------------------------------------------------------------------|------------------------------|
| bootstrapServers=ip1:port1,ip2:port2,ip3:port3 | Kafka 连接信息按需配置 |
| group.id=getendpos | auto.commit.interval.ms=1000 |
| autoOffsetReset=earliest | 建议保持默认 |
| enableAutoCommit=false | 建议保持默认 |
| autoCommitIntervalMs=1000 | 建议保持默认 |
| sessionTimeoutMs=30000 | 建议保持默认 |
| maxPartitionFetchBytes=10485760 | 建议保持默认 |
| maxPollRecords=10 | 建议保持默认 |
| maxPollIntervalMs=300000 | 建议保持默认 |
| key.deserializer=org.apache.kafka.common.s erialization.StringDeserializer | 必须保持默认 |

| 配置参数 | 说明 |
|--------------------------------------------|---------|
| value.deserializer=org.apache.kafka.common | s必须保持默认 |
| erialization.ByteArrayDeserializer | |

- 待所有配置修改并确认无误后, 启动程序即可。
 - ■启动命令:

```
./xgsyn.sh start
```

■ 查看是否启动同步软件命令:

```
./xgsyn.sh status
```

■ 停止同步软件命令:

```
./xgsyn.sh stop
```

4.3 Kafka 同步至数据库

4.3.1 使用 Web 端配置启动

以 web 形式启动命令:

```
./xgsyn.sh start\_web
```

Kafka 为源端同步至数据库,需要在源端类型选择"Kafka",目标端类型选择"库",并配置同步软件配置信息和数据源配置信息。

• 同步软件配置信息如下所示。



图 4-14 同步配置信息

表 4-15 源端 kafka 配置

| 配置参数 | 说明 |
|---------------------|------------------------------|
| sourceKafkaParath=8 | 每个 topic 的分区数 |
| sourceTopic | 需要同步的 topic 名,以"\\"分隔。 |
| | 例:test.d2.tclob \\test.u2.t1 |

● 消费者配置信息如表4-16所示。

表 4-16 consumer.properties 消费者配置

| 配置参数 | 说明 |
|----------------------------------------------|---------------------|
| bootstrapServers=ip1:port1,ip2:port2,ip3:por | Kafka 连接信息按需配置 |
| t3 | |
| groupId=getendpos | 若同步工具配置页面的正向同步过滤选项为 |
| | 是,则是选择需要同步的,为否则是选择不 |
| | 需要同步的 |
| autoOffsetReset=earliest | 建议保持默认 |
| enableAutoCommit=false | 建议保持默认 |
| autoCommitIntervalMs=1000 | 建议保持默认 |
| sessionTimeoutMs=30000 | 建议保持默认 |
| maxPartitionFetchBytes=10485760 | 建议保持默认 |
| maxPollRecords=10 | 建议保持默认 |
| maxPollIntervalMs=300000 | 建议保持默认 |
| key.deserializer=org.apache.kafka.common.s | 必须保持默认 |
| erialization.StringDeserializer | |

| 配置参数 | 说明 |
|--------------------------------------------|----------|
| value.deserializer=org.apache.kafka.common | .s必须保持默认 |
| erialization.ByteArrayDeserializer | |

● 生产者配置信息如表4-17所示。

表 4-17 producer.properties 生产者配置

| 配置参数 | 说明 |
|------------------------------------------------|----------------|
| bootstrapServers=ip1:port1,ip2:port2,ip3:por | Kafka 连接信息按需配置 |
| t3 | |
| compressionType=none | 建议保持默认 |
| lingerMs=10 | 建议保持默认 |
| acks=all | 建议保持默认 |
| retries=8 | 建议保持默认 |
| batchSize=163840 | 建议保持默认 |
| bufferMemory=33554432 | 建议保持默认 |
| enableIdempotence=true | 建议保持默认 |
| key.serializer=org.apache.kafka.common.se | 必须保持默认 |
| rialization.StringSerializer | |
| value.serializer=org.apache.kafka.common.s | 必须保持默认 |
| erialization.ByteArraySerializer | |
| partitioner.class=com.sync.util.BinlogPartitio | 必须保持默认 |
| ner | |

山 说明

按照实际部署情况,将 bootstrap.servers 修改为 Kafka 集群的 IP 和端口,其余参数保持默认即可。

● 数据源配置信息如图 4-15 所示。



图 4-15 数据源配置信息

表 4-18 目标库配置

| 配置参数 | 说明 | |
|------------|-----------------------------|--|
| 目标库数据库类型 | 目标库类型(xugu,oracle,mysql) | |
| 模式名 | 目标端模式名,严格区分大小写 | |
| 目标库名 | 目标库库名 | |
| 用户名 | 目标端用户名 | |
| 密码 | 目标端密码(web 端使用明文密码) | |
| ip 地址 | 目标端数据库 IP 地址 | |
| 端口 | 目标端数据库端口号 | |
| 同步时写入线程数 | 同步时写入线程数,需要与 Binlog 分区数对应一致 | |
| 每个写入线程的并发数 | 入库并发数 | |

4.3.2 使用 jar 包配置启动

以 Kafka 为源端同步至数据库,仅能解析由虚谷 Binlog 的变更组成的 kafka 消息队列。除了 cfg.properties 基础配置需要修改之外,还需要修改 Kafka 的消费者和生产者,对应的配置文件 分别为 consumer.properties 和 producer.properties。还需要配置目标端数据库信息 dbcfg.properties。

● cfg.properties 具体配置如表4-19所示。

表 4-19 cfg.properties 具体配置

| 配置参数 | 说明 |
|-----------------------------|---------------------------------------------------------------------|
| #type in (db,kafka,file) | - |
| source.type=db | 源端类型 |
| aim.type=db | 目标端类型 |
| #binlog source config | 源端数据库配置信息 |
| source.log.db=db | 待同步库名 |
| source.db.usr=usr_sod | 源端用户名 |
| source.db.pwd=password | 源端用户名密码(通过 Encryption.jar 加密后的密码) |
| source.db.port=12345 | 源端端口号 |
| source.log.suber.name=sync2 | 同步使用的订阅名 |
| source.log.filter.T=null | 同步过滤表达式,过滤指定表。格式:模式 名:表名,多个表之间以&符号分割。例: s1:t1&s2:t2。(均使用英文字符) |
| source.log.filter.V=null | 同步过滤表达式, 过滤指定视图 |
| source.log.filter.P=null | 同步过滤表达式, 过滤指定存储过程 |
| source.log.filter.F=null | 同步过滤表达式, 过滤指定存储函数 |

| 配置参数 | 说明 | |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------|--|
| source.log.forward=true | 与 source.log.filter 配合使用,若为 true,则同步 source.log.filter 中所配置的表。若为 false,则不同步 source.log.filter 中所配置的表。默认为 true | |
| source.log.partion=8 | 虚谷数据库日志分区 | |
| source.log.fetch.size=20 | 同步每次拉取日志大小 | |
| source.log.ips=127.0.0.1 | 单事务最大行数 | |
| source.commit_rows=1000 | 同步过滤表达式, 过滤指定存储函数 | |
| ddl.support=false | 是否进行 DDL 同步 | |
| redo.offset.server=true | 是否启用数据库端订阅者偏移量 | |
| reconnect.num=-1 | 数据库重连次数,为-1 则为无限重连 | |
| disable.binlog=false | true 目标端 binlog 不记录同步的数据,false 为记录 | |
| isMax.binlog=false | false 从 binlog 原始位置开始读取,true 从最新的文件号的 0 偏移量开始读取 | |
| # kafka source config | 源端 kafka 配置 | |
| source.kafka.parath=8 | 每个 topic 的分区数 | |
| source.topic=test.d2.tclob\\test.u2.t1 | 需要同步的 topic 名,以"\\"分隔 | |
| #to kafka | 写入 Kafka 消息中间件配置 | |
| zk.connect=192.168.2.225:2181 | zookeeper ip | |
| topic.partition=8 | Kafka 每个 topic 的分区数 | |
| repeat.pos.kafka=true | 是否依赖 Kafka 数据进行断点续传过滤重复 | |

| 配置参数 | 说明 | |
|----------------------|--------------------------------------------------------------------------------------------|--|
| case.sensitive=true | 用于同步至 kafka 时创建 topic 的大小写控制。若为 true,则创建的 topic 名大小写按照源端库实际大小写创建。若为 false,则创建的 topic 名全为小写 | |
| #to db | 写入数据库 | |
| writer.thd.num=8 | 同步时写入线程数(建议和 source.log.partion相同) | |
| writer.db=xugu | 目标库类型(xugu, oracle, mysql) | |
| writer.schema=SYSDBA | 指定目标端模式名(严格区分大小写) | |
| writer.parathd.num=1 | 入库并发数 | |
| # to Http | El 信息 | |
| url.ei= | 针对天境 EI,可不填 | |
| task.id= | 针对天境 EI,可不填 | |
| task.name= | 针对天境 EI,可不填 | |

山 说明

按照实际部署情况,将 bootstrap.servers 修改为 Kafka 集群的 IP 和端口,其余参数保持默认即可。

● consumer.properties 消费者配置如表4-20所示。

表 4-20 consumer.properties 消费者配置

| 配置参数 | 说明 |
|-------------------------------------------------------------------------------|-----------------------------------------------------|
| bootstrapServers=ip1:port1,ip2:port2,ip3:port3 | Kafka 连接信息按需配置 |
| groupId=getendpos | 若同步工具配置页面的正向同步过滤选项为 是,则是选择需要同步的,为否则是选择不 需要同步的 |
| autoOffsetReset=earliest | 建议保持默认 |
| enableAutoCommit=false | 建议保持默认 |
| autoCommitIntervalMs=1000 | 建议保持默认 |
| sessionTimeoutMs=30000 | 建议保持默认 |
| maxPartitionFetchBytes=10485760 | 建议保持默认 |
| maxPollRecords=10 | 建议保持默认 |
| maxPollIntervalMs=300000 | 建议保持默认 |
| key.deserializer=org.apache.kafka.common.s erialization.StringDeserializer | 必须保持默认 |
| value.deserializer=org.apache.kafka.common erialization.ByteArrayDeserializer | .s必须保持默认 |

- dbcfg.properties 只需要配置目标端数据库连接信息。
- 同步至虚谷数据库,如表4-21所示。

表 4-21 dbcfg.properties 详细配置(虚谷)

| 库信息 | 配置参数 | 说明 |
|---------|-----------------------|----|
| 目标库连接信息 | target-db.user=SYSDBA | - |

| 库信息 | 配置参数 | 说明 |
|---------|----------------------------------------------------------------|------------------------------|
| 目标库连接信息 | target-db.password=SYSDBA | 通过 Encryption.jar 加密后的 密码 |
| 目标库连接信息 | target- db.driverClass=com.xugu.clou djdbc.Driver | - |
| 目标库连接信息 | target- db.jdbcUrl=jdbc:xugu://1 9 2 .168.2.222:9930/dcc | - |

● 同步至 MySQL 数据库,如表4-22所示。

表 4-22 dbcfg.properties 详细配置(MySQL)

| 库信息 | 配置参数 | 说明 |
|---------|-------------------------------------------------------------------------------------------------|------------------------------|
| 目标库连接信息 | target-db.user=SYSDBA | - |
| 目标库连接信息 | target-db.password=SYSDBA | 通过 Encryption.jar 加密后的 密码 |
| 目标库连接信息 | target- db.driverClass=com.mysql.cj.j dbc.Driver | - |
| 目标库连接信息 | target- db.jdbcUrl=jdbc:mysql://192.1 68.2.85:3306/d2?useUnicode =true&characterEncoding=utf- 8 | - |

■ 同步至 Oracle 数据库,如表4-23所示。

表 4-23 dbcfg.properties 详细配置 (Oracle)

| 库信息 | 配置参数 | 说明 |
|---------|------------------------------------------------------------------|------------------------------|
| 目标库连接信息 | target-db.user=u2 | - |
| 目标库连接信息 | target- db.password=MTlzNDU2 | 通过 Encryption.jar 加密后的 密码 |
| 目标库连接信息 | target- db.driverClass=oracle.jdbc.dri ver.OracleDriver | - |
| 目标库连接信息 | target-db.jdbcUrl=jdbc:oracle:t hin:@//192.168.2.17:1521/orcl | - |

待所有配置修改并确认无误后, 启动程序即可。

○启动命令:

./xgsyn.sh start

• 查看是否启动同步软件命令:

./xgsyn.sh status

· 停止同步软件命令:

./xgsyn.sh stop

4.3.3 校验程序配置

同步软件程序包中还提供了一个 Kafka 数据校验程序,此程序主要用于从 Kafka 拉取日志中的 SQL 语句,配置文件为 poll.properties,配置如表4-24所示。

表 4-24 poll.properties 配置

| 配置参数 | 说明 |
|------------------------------------------------|----------------|
| bootstrapServers=ip1:port1,ip2:port2,ip3:port3 | Kafka 连接信息按需配置 |
| group.id=pollsql | 建议保持默认 |
| auto.offset.reset=earliest | 建议保持默认 |
| enable.auto.commit=false | 建议保持默认 |
| autoCommitIntervalMs=1000 | 建议保持默认 |
| session.timeout.ms=30000 | 建议保持默认 |
| max.partition.fetch.bytes=10485760 | 建议保持默认 |
| max.poll.records=10 | 建议保持默认 |
| max.poll.interval.ms=300000 | 建议保持默认 |
| key.deserializer=org.apache.kafka.common.s | 必须保持默认 |
| erialization.StringDeserializer | |
| value.deserializer=org.apache.kafka.common | s必须保持默认 |
| erialization.ByteArrayDeserializer | |

按照实际部署情况,将 bootstrap.servers 修改为 Kafka 集群的 IP 和端口,其余参数默认即可。配置完成后即可启动 Kafka 校验程序,程序执行时需要两个参数:

- 参数 1: topic 名称
- 参数 2: topic 消息偏移量
- 拉取数据直接在控制台输出:

java -jar PollBinlog.jar 参数1 参数2

● 拉取数据写入目标文件 sql.txt:

java -jar PollBinlog.jar 参数1 参数2 > sql.txt

4.3.4 主备配置

为避免服务器硬件异常导致同步软件停止并无法重启,所以提供主备配置方案。 提供两台相同服务器,均按照文档安装与配置部分搭建工具。备份同步软件配置文件 cfg.properties 中 redo.offset.server 需设置为 true,其他配置项均主备相同。当主服务器不可 使用后,在备用服务器启动同步软件即可。

5 日常运维

5.1 binlog 日志管理

5.1.1 开启 binlog 日志

启用语法:

● 库级

```
DBMS\_REPLICATION.CREATE\_MODIFY\_SOURCE('*','*');
```

● 模式级

```
DBMS\_REPLICATION.CREATE\_MODIFY\_SOURCE('schemaname','*');
```

● 表级

```
DBMS\_REPLICATION.CREATE\_MODIFY\_SOURCE('schemaname','
    tablename');
```

在对象上启用备份后相应对象系统表(dba_databases、dba_schemas、dba_tables)REG_MODIFY 字段标记为 T。

通过虚谷数据库管理工具或控制台工具登录指定库,按需通过以上启用语法启用对象 binlog 记载,如图5-1所示。

5.1.2 关闭 binlog 日志

关闭语法:

库级

```
DBMS\_REPLICATION.DROP\_MODIFY\_SOURCE ('*','*');
```

● 模式级

```
DBMS\_REPLICATION.DROP\_MODIFY\_SOURCE ('schemaname','*');
```

表级

```
DBMS\_REPLICATION.DROP\_MODIFY\_SOURCE ('schemaname','
  tablename');
```

在对象上关闭备份后相应对象系统表(dba_databases、dba_schemas、dba_tables)REG_MODIFY 字段标记为 F。

```
SQL> exec dbms_replication.create_modify_source('*','*');
Execute successful.
Use time:315 ms.
SQL> select db_name,reg_modify from dba_databases where db_name='DB_TEST';
DB_NAME | REG_MODIFY |
DB_TEST| T |
Total 1 records.
Use time:5 ms.
SOL> exec dbms replication.create modify source('u test','*');
Execute successful.
SQL> select schema_name,reg_modify from dba_schemas where schema_name='U_TEST';
SCHEMA_NAME | REG_MODIFY |
U_TEST| T |
Total 1 records.
Use time:21 ms.
SQL> exec dbms_replication.create_modify_source('u_test','t_test');
Execute successful.
Use time:23 ms.
SQL> select table_name,reg_modify from dba_tables where table_name='T_TEST';
TABLE_NAME | REG_MODIFY |
T_TEST| T |
```

图 5-1 开启 binlog 日志

通过虚谷数据库管理工具或控制台工具登录指定库,按需通过以上关闭语法关闭对象 binlog 记载,如图5-2所示。

5.1.3 查询 binlog 日志

虚谷数据库以系统函数接口的方式,向用户提供 binlog 日志查询。binlog 日志必须通过订阅者(订阅者的相关操作见章节 6.2 订阅管理)绑定的用户登录至 binlog 日志记载节点 (G 角色所在节点) 才能进行查询,并且查询的结果以结果集的方式返回(返回可执行的 SQL 语句)。查询函数为:

```
DBMS\_REPLICATION.POLL\_MODIFY\_DATA
```

该函数参数及含义如表5-1所示。

表 5-1 DBMS_REPLICATION.POLL_MODIFY_DATA 参数说明

| 参数名称 | 参数类型 | 说明 |
|-----------------|---------|----------------------|
| SUBSCRIBER_NAME | VARCHAR | 订阅名 |
| PARTI_NO | INTEGER | 分区号(0 到 mod- |
| | | ify_log_parti_num-1) |

| 参数名称 | 参数类型 | 说明 |
|----------|---------|----------------------------|
| FILTER | VARCHAR | 过滤条件 |
| POLL_LEN | BIGINT | 本次查询需要扫描的日志文件长度(字节) |
| KEEP_POS | BOOLEAN | 是否移动当前分区的备份数据读取位置到本次读取结束位置 |
| FNO | INTEGER | 本次查询开始的文件号 |
| FPOS | BIGINT | 本次查询的开始文件内偏移量 |

示例:

```
exec dbms\_replication.poll\_modify\_data('test',0,'obj\_id=1048577
',10*1024*1024,true,0,0);
```

本例表示本次查询 0 号文件, 偏移 0 个字节开始扫描, 扫描长度为 10M, 并且只查询对象 ID 为 1048577 的日志, 查询结果如图5-3所示。

以上查询是通过控制台执行的,因此在结果集不多的情况下,可以非常方便的观察查询结果。 在无法预知结果集的情况下,建议将查询结果集重定向到文件,方便后续的分析与使用,操作 步骤如下 (Linux 下执行):

•新建一个脚本文件 expbinlog.sh, 并赋予执行权限。

```
touch expbinlog.sh \&\& chmod +x expbinlog.sh
```

• 使用 vim 编辑 expbinlog.sh 脚本,新增以下内容。

根据实际的使用情况来修改 IP、端口、库名、用户名和密码以及查询参数,修改完成后保存退出,然后执行以下命令:

```
./expbinlog.sh >> binlog\_x\_y.txt
```

建议将 x 设置为对象 ID 号, y 设置为当前查询的分区号(查询函数中的第2个参数)。

```
SQL> exec dbms_replication.drop_modify_source('*','*');
Execute successful.
Use time:19 ms.
SQL> select db_name, reg_modify from dba_databases where db_name='DB_TEST';
DB_NAME | REG_MODIFY |
DB_TEST| F |
Total 1 records.
Use time:7 ms.
SQL> exec dbms_replication.drop_modify_source('u_test','*');
Execute successful.
Use time:26 ms.
SQL> select schema_name,reg_modify from dba_schemas where schema_name='U_TEST';
SCHEMA_NAME | REG_MODIFY |
U_TEST| F |
Total 1 records.
Use time:23 ms.
{\tt SQL} \verb|> exec dbms_replication.drop_modify_source('u_test', 't_test');}
Use time:170 ms.
SQL> select table_name,reg_modify from dba_tables where table_name='T_TEST';
TABLE_NAME | REG_MODIFY |
T_TEST| F |
```

图 5-2 关闭 binlog 日志

图 5-3 查询结果

5.2 订阅管理

5.2.1 创建订阅者

订阅者创建语法:

```
DBMS\ REPLICATION.CREATE\ SUBSCRIBER('suberName', 'username');
```

```
SQL> exec dbms_replication.create_subscriber('s_test1','u_test');

Execute successful.
Use time:24 ms.

SQL> exec dbms_replication.create_subscriber('s_test2','u_test');

Execute successful.
Use time:114 ms.

SQL> select ds.subs_name,du.user_name from dba_subscribers ds
2 join dba_users du using(user_id) where du.user_name='U_TEST';

SUBS_NAME | USER_NAME |

s_test1| U_TEST|
s_test2| U_TEST|

S_test2| U_TEST|

Total 2 records.
```

图 5-4 创建订阅者

5.2.2 删除订阅者

订阅者删除语法:

```
DBMS\_REPLICATION.DROP\_SUBSCRIBER ('suberName', 'username');
```

如下图所示(本例删除上一步为用户 U TEST 创建的订阅者)。

```
SQL> exec dbms_replication.drop_subscriber('s_test1','u_test');

Execute successful.
Use time:463 ms.

SQL> exec dbms_replication.drop_subscriber('s_test2','u_test');

Execute successful.
Use time:355 ms.

SQL> select ds.subs_name,du.user_name from dba_subscribers ds
2 join dba_users du using(user_id) where du.user_name='U_TEST';

SUBS_NAME | USER_NAME |

Total 0 records.
```

图 5-5 删除订阅者

5.3 消费查询

5.3.1 系统表查询

虚谷数据库同步软件以消费者方式从虚谷数据库 G 节点消费数据,发送到 Kafka 消费进度查看方式:

```
select * from all\ streams order by parti\ no;
```

```
### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_no;

#### SQL\> select * from all_streams order by parti_
```

图 5-6 查看消费进度

观测 8 个分区的 file off 是否正常推进。

查看当前数据库文件写入位置:

```
select * from dba\_modify\_logs;
```

5.3.2 同步软件日志查询

```
tail -flog/info.log
```

确认 part x (x 为 0 至 7) 均输出同步消息。

```
DB_ID | PARTI_NO | SRV_NID | MIN_FILE_NO | MAX_FILE_NO | RESERVED1 | RESERVED2 |

2 | 1 | CNULL | 0 | 245 | CNULL | CNULL |
2 | 3 | CNULL | 0 | 245 | CNULL | CNULL |
2 | 0 | CNULL | 0 | 245 | CNULL | CNULL |
2 | 7 | CNULL | 0 | 245 | CNULL | CNULL |
2 | 7 | CNULL | 0 | 245 | CNULL | CNULL |
2 | 7 | CNULL | 0 | 245 | CNULL | CNULL |
2 | 2 | CNULL | 0 | 245 | CNULL | CNULL |
2 | 4 | CNULL | 0 | 245 | CNULL | CNULL |
2 | 6 | CNULL | 0 | 245 | CNULL | CNULL |
3 | CNULL | 0 | 245 | CNULL | CNULL |
4 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNULL |
5 | CNULL | 0 | 245 | CNUL
```

图 5-7 查看写入位置

```
runinfo] 20199623 10:09:27 - sny_log part 3, rows 1745069 - couninfo] 20199623 10:09:27 - sny_log part 1, rows 1745069 - couninfo] 20199623 10:09:27 - sny_log part 1, rows 1745069 - couninfo] 20199623 10:09:27 - sny_log part 1, rows 1745069 - couninfo] 20199623 10:09:27 - sny_log part 2, rows 1745070 - couninfo] 20199623 10:09:20 - sny_log part 3, rows 2, 237, 55123434 - couninfo] 20199623 10:09:20 - sny_log part 2, rows 1745070 - couninfo] 20199623 10:09:20 - sny_log part 2, rows 172612 post 2, 237, 46534355 - couninfo] 20199623 10:09:20 - sny_log part 2, rows 172612 post 2, 237, 55743516 - couninfo] 20199623 10:09:20 - sny_log part 2, rows 172612 post 2, 237, 55743516 - couninfo] 20199623 10:09:20 - sny_log part 2, rows 172612 post 2, 237, 55743516 - couninfo] 20199623 10:09:20 - sny_log part 2, rows 172612 post 2, 237, 55743516 - couninfo] 20199623 10:09:20 - sny_log part 2, rows 172612 post 3, 237, 56129600 - sny_log part 3, rows 1751084 post; 1, 237, 75302002 - sny_log part 1, rows 1753084 post; 1, 237, 75302002 - sny_log part 3, rows 1751084 post; 2, 237, 4045503033 - sny_log part 3, rows 1751084 post; 2, 237, 4045503033 - sny_log part 3, rows 1751084 post; 3, 237, 604199053 - sny_log part 3, rows 1751084 post; 3, 237, 604199053 - sny_log part 3, rows 1751084 post; 3, 237, 604199053 - sny_log part 3, rows 1751084 post; 3, 237, 604199053 - sny_log part 3, rows 1751084 post; 3, 237, 604199053 - sny_log part 3, rows 1751084 post; 3, 237, 604199053 - sny_log part 3, rows 1751084 post; 3, 237, 604199053 - sny_log part 3, rows 1751084 post; 2, 237, 4042600177 - sny_log part 4, rows 1634040 post; 4, 236, 10651051572 - sny_log part 4, rows 1634040 post; 4, 236, 10651051583 - sny_log part 4, rows 1634040 post; 4, 236, 10651051583 - sny_log part 4, rows 1634040 post; 4, 236, 10651051583 - sny_log part 4, rows 1634040 post; 4, 236, 10651051583 - sny_log part 4, rows 1634040 post; 4, 236, 10651051583 - sny_log part 4, rows 1634040 post; 4, 236, 10651051583 - sny_log part 4, rows 1634040 post; 4, 236, 10651051583 - sn
```

图 5-8 同步软件日志查询

5.3.3 Kafka 查询

进入 Kafka 安装目录。

执行:

bin/kafka-consumer-groups.sh --bootstrap-server ip:9092 --list

```
linus -p2s8:/home/sdc/kafka_2.11-2.1.0 # bin/kafka-consumer-groups.sh --bootstrap-server 10.40.17.64:9092 --li st console-consumer-82000 music-group-2 HADB consolumer-47401 getendops STBD di-receiver Linus-p2s8:/home/sdc/kafka_2.11-2.1.0 # |
```

图 5-9 Kafka 查询 1

执行:

bin/kafka-consumer-groups.sh --bootstrap-server ip:9092 --group 消费组名 --describe

- current_offset: 当前已消费数
- Log-end-offset: topic 分区总消息数据
- Lag: 未消费数据
 - 为 0 表示已消费完成
 - 在增大证明虚谷数据库同步软件推送数据到 Kafka 速率大于 Kafka 数据被消费速率
 - 在减少证明推送至 Kafka 的数据在被正常消费

执行:

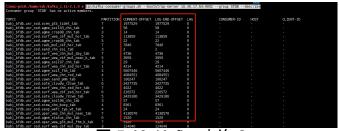


图 5-10 Kafka 查询 2

bin/kafka-consumer-groups.sh --bootstrap-server ip:9092 --group 消费组名 --describe |sort -k 1 |grep 小写表名

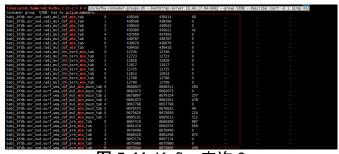


图 5-11 Kafka 查询 3

主要查看指定表名消费情况: Log-end-offset current_offset Lag

5.3.4 Kafka 数据验证

进入虚谷数据库同步软件安装目录。

执行:

java -jar pollbinlog.jar 参数1 参数2 > 输出文件

该命令从 Kafka 拉取数据。

参数说明

- 参数 1: topic 名字
- 参数 2: 起始偏移量 (不设置默认从头开始拉取)
- 输出文件: 自行指定重定向到某文件

5.3.5 Kafka 偏移推进

进入 Kafka 安装目录。

执行:

bin/kafka-consumer-groups.sh --bootstrap-server ip:9092 --group 消费组名 --reset-offsets --topic babj_bfdb.usr_sod.tabname:0 --to-offset 具体偏移 --execute

5.4 常见问题

5.4.1 启动报错

用户初次使用虚谷数据库同步软件时,可能会出现启动报错。此错误是由于初次使用时,数据库系统中还未产生 binlog 日志(初次开启 binlog 日志记载),启动时,同步软件无法找到相应的日志文件,因此报错。

出现此错误时,需要使用同步软件中配置的订阅者绑定的用户,登录到相应的库下,执行任意 binglog 日志记载支持的操作,产生 binlog 日志。然后重新启动同步软件即可。

5.4.2 线程异常

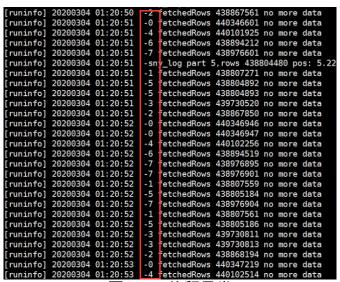


图 5-12 线程异常

查看 log/info.log,通过命令 **tailf info.log** 查看文件刷新内容,分区号 0 到 7 是否正常刷新出现。若有分区号未出现,则重启同步软件。



成都虚谷伟业科技有限公司

联系电话: 400-8886236

官方网站: www.xugudb.com