

云数据迁移

# 常见问题

文档版本 19  
发布日期 2024-12-05



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 安全声明

## 漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

# 目录

|  |           |
|--|-----------|
| <b>1 通用类</b>                               | <b>1</b>  |
| 1.1 CDM 与其他数据迁移服务有什么区别，如何选择？               | 1         |
| 1.2 CDM 有哪些优势？                             | 4         |
| 1.3 CDM 有哪些安全防护？                           | 5         |
| 1.4 如何降低 CDM 使用成本？                         | 5         |
| 1.5 CDM 未使用数据传输功能时，是否会计费？                  | 6         |
| 1.6 已购买包年包月的 CDM 套餐包，为什么还会产生按需计费的费用？       | 6         |
| 1.7 如何查看套餐包的剩余时长？                          | 6         |
| 1.8 套餐包到期未续订或按需资源欠费时，我的数据会保留吗？             | 7         |
| 1.9 CDM 可以跨账户使用吗？                          | 7         |
| 1.10 CDM 集群是否支持升级操作？                       | 7         |
| 1.11 CDM 迁移性能如何？                           | 7         |
| 1.12 CDM 不同集群规格对应并发的作业数是多少？                | 8         |
| 1.13 CDM 集群可以关机吗？                          | 9         |
| <b>2 功能类</b>                               | <b>10</b> |
| 2.1 是否支持增量迁移？                              | 10        |
| 2.2 是否支持字段转换？                              | 10        |
| 2.3 Hadoop 类型的数据源进行数据迁移时，建议使用的组件版本有哪些？     | 17        |
| 2.4 数据源为 Hive 时支持哪些数据格式？                   | 18        |
| 2.5 是否支持同步作业到其他集群？                         | 18        |
| 2.6 是否支持批量创建作业？                            | 18        |
| 2.7 是否支持批量调度作业？                            | 19        |
| 2.8 如何备份 CDM 作业？                           | 19        |
| 2.9 如何解决 HANA 集群只有部分节点和 CDM 集群网络互通？        | 19        |
| 2.10 如何使用 Java 调用 CDM 的 Rest API 创建数据迁移作业？ | 20        |
| 2.11 如何将云下内网或第三方云上的私网与 CDM 连通？             | 25        |
| 2.12 CDM 是否支持参数或者变量？                       | 27        |
| 2.13 CDM 迁移作业的抽取并发数应该如何设置？                 | 27        |
| 2.14 CDM 是否支持动态数据实时迁移功能？                   | 29        |
| 2.15 CDM 是否支持集群关机功能？                       | 29        |
| 2.16 如何使用表达式方式获取当前时间？                      | 30        |
| 2.17 在创建迁移作业时，where 语句参数中的时间格式是怎样的？        | 30        |
| 2.18 CDM 作业可以将源表中的字段注释迁移到目标端表吗？            | 30        |

|  |           |
|--|-----------|
| 2.19 如何查询数据连接创建人? .....  | 30        |
| <b>3 故障处理类.....</b>  | <b>32</b> |
| 3.1 日志提示解析日期格式失败时怎么处理? .....   | 32        |
| 3.2 字段映射界面无法显示所有列怎么处理? .....   | 34        |
| 3.3 CDM 迁移数据到 DWS 时如何选取分布列? .....  | 38        |
| 3.4 迁移到 DWS 时出现 value too long for type character varying 怎么处理? .....                                      | 39        |
| 3.5 OBS 导入数据到 SQL Server 时出现 Unable to execute the SQL statement 怎么处理? .....                               | 40        |
| 3.6 获取集群列表为空/没有权限访问/操作时报当前策略不允许执行? .....   | 41        |
| 3.7 Oracle 迁移到 DWS 报错 ORA-01555.....   | 42        |
| 3.8 MongoDB 连接迁移失败时如何处理? .....   | 43        |
| 3.9 Hive 迁移作业长时间卡顿怎么办? .....   | 43        |
| 3.10 使用 CDM 迁移数据由于字段类型映射不匹配导致报错怎么处理? .....   | 43        |
| 3.11 MySQL 迁移时报错“JDBC 连接超时”怎么办? .....  | 44        |
| 3.12 创建了 Hive 到 DWS 类型的连接, 进行 CDM 传输任务失败时如何处理? .....   | 45        |
| 3.13 如何使用 CDM 服务将 MySQL 的数据导出成 SQL 文件, 然后上传到 OBS 桶? .....  | 45        |
| 3.14 如何处理 CDM 从 OBS 迁移数据到 DLI 出现迁移中断失败的问题? .....   | 46        |
| 3.15 报错“配置项[linkConfig.createBackendLinks]不存在”或“配置项<br>[throttlingConfig.concurrentSubJobs] 不存在怎么办”? ..... | 46        |
| 3.16 新建 MRS Hive 连接时, 提示: CORE_0031:Connect time out. (Cdm.0523) 怎么解决? .....                               | 46        |
| 3.17 迁移时已选择表不存在时自动创表, 提示“CDM not support auto create empty table with no<br>column”怎么处理? .....             | 47        |
| 3.18 创建 Oracle 关系型数据库迁移作业时, 无法获取模式名怎么处理? .....   | 47        |
| 3.19 MySQL 迁移时报错: invalid input syntax for integer: "true".....  | 48        |
| 3.20 作业源端是 Oracle 时, 运行时间过长报 snapshot too old 怎么解决? .....  | 48        |
| 3.21 整库迁移到 Hive, 报错 Identifier name is too long 如何处理? .....  | 49        |
| 3.22 迁移数据到 DLI 时有数据丢失怎么处理? .....   | 49        |
| 3.23 创建 Oracle 数据连接测试连通性成功, 连接管理界面中测试连接失败。是什么原因? .....   | 50        |
| 3.24 作业配置表不存在时自动创建, 目的端字段映射不出来怎么处理? .....  | 51        |
| 3.25 作业从旧集群导出, 再导入到新的集群失败怎么解决? .....   | 52        |
| 3.26 迁移 HDFS 文件, 报错无法获取块怎么处理? .....  | 53        |
| 3.27 创建 MRS 数据连接时测试连通性报下载配置文件失败错误怎么处理? .....   | 53        |
| 3.28 CDM 作业管理访问不了, 提示网络或服务器访问异常怎么处理? .....   | 54        |
| 3.29 通过 CDM 从 OBS 迁移数据到 DLI, 同样的作业在新版本集群迁移失败? .....  | 54        |
| 3.30 CDM 迁移 DWS 数据报错 Read timedout 怎么处理? .....   | 55        |
| 3.31 CDM 集群 Hive 连接无法查询库和表的内容.....   | 56        |
| 3.32 创建 FusionInsight HDFS 连接报错 get filesystem 怎么解决? .....   | 57        |
| 3.33 Mysql 导入数据到 DLI, 快执行完时失败了提示 Invoke DLI service api failed 错误怎么解决? .....                               | 59        |
| 3.34 作业配置添加字段, MongoDB 字段映射存在问题.....   | 60        |
| 3.35 DLI 外表(OBS 文件)迁移 DWS 某字段转义, 带有“\” .....   | 63        |
| 3.36 执行 Postgresql-to-Hive 迁移作业报错“Error occurs during loader run” .....                                    | 64        |
| 3.37 迁移 Mysql 到 DWS 报错“Lost connection to MySQL server during query”怎么处理? .....                            | 66        |
| 3.38 迁移 MySql 到 DLI 字段类型转换报错 For input string: "false"怎么处理? .....  | 68        |
| 3.39 迁移 MySql 到 DWS, TINYINT 类型迁移报错.....   | 70        |

|   |    |
|---|----|
| 3.40 数据迁移前后数据量不一致是什么问题? .....   | 72 |
| 3.41 创建源数据连接, 一直报错用户名和密码错误, 但是实际填的没有错.....                              | 73 |
| 3.42 数据库写入 OBS 场景, 表中小驼峰命名字段, 提示字段不存在.....                              | 74 |
| 3.43 CSV 数据类型插入 MySQL 报错 invalid utf-8 charater string " .....          | 74 |
| 3.44 定时任务失败, 检查连接器连接存在问题.....   | 74 |
| 3.45 脏数据导致 CSV 数据类型问题插入 MySQL 报错.....                                   | 75 |
| 3.46 写 ES 报 timeout waiting for connection from pool 错误怎么解决? .....      | 76 |
| 3.47 Oracle 迁移到 DWS 报错 ORA-01555.....                                   | 77 |
| 3.48 FTP 测试连通性失败, 报服务器内部错误怎么解决? .....                                   | 77 |
| 3.49 CDM 连接 RDS-Mysql, 除 root 用户外, 其他用户都报错, 怎么办? .....                  | 78 |
| 3.50 MRS Hive, MRS Kafka, MRS Hudi 数据源创建连接时 IP 长度校验不通过, 如何处理? .....     | 79 |
| 3.51 CDM 转换空字符串表达式 StringUtils.defaultIfEmpty 不生效怎么解决? .....            | 81 |
| 3.52 当 Hudi 表属性和 hoodie.properties 文件中分区信息不一致时, 会导致数据写入失败怎么解决? .....    | 82 |
| 3.53 当 MySQL、Oracle、PostgreSQL 作为源端时, 如果作业报错“Read timed out”怎么处理? ..... | 82 |
| 3.54 RDS-Mysql 和开源 Mysql 对于 date 类型的范围不一致, 可能导致 date 数据转换异常.....        | 82 |
| 3.55 通过 json 方式导入的作业, sql server 迁移到 hive, 执行作业报错.....                  | 83 |
| 3.56 Hudi 源端的作业长时间处于 BOOTING 状态怎么解决? .....                              | 84 |
| 3.57 Hudi 源端作业, 读 Hudi 作业字段映射多了一列 col, 作业执行失败怎么处理? .....                | 85 |
| 3.58 Hudi 目的端的作业自动建表报错: schema 不匹配, 建表失败怎么办? .....                      | 85 |
| 3.59 Hudi 目的端的作业失败, 日志报错 Read Timeout 怎么解决? .....                       | 86 |
| 3.60 Hudi 目的端的作业执行卡 Running, 读取行数写入行数相等且不再增加怎么解决? .....                 | 87 |
| 3.61 Hudi 目的端的作业执行卡 Running, 数据写入行数一直显示为 0 如何处理? .....                  | 88 |
| 3.62 Hudi 目的端的作业执行 Spark SQL 写入 Hudi 失败怎么办? .....                       | 89 |
| 3.63 Hudi 目的端的作业执行过程中, 由于源端连接闪断、超时或主动终止连接导致作业执行失败怎么处理? .....            | 92 |

# 1 通用类

## 1.1 CDM 与其他数据迁移服务有什么区别，如何选择？

华为云上涉及数据迁移的服务有以下几种：

- [云数据迁移服务 CDM](#)
- [对象存储迁移服务 OMS](#)
- [数据复制服务 DRS](#)
- [主机迁移服务 SMS](#)
- [数据库和应用迁移 UGO](#)
- [数据快递服务 DES](#)

上述数据迁移服务的区别请参见[各个数据迁移服务区别](#)。

### 什么是云数据迁移服务(CDM)？

云数据迁移（Cloud Data Migration，简称CDM）是一种高效、易用的数据集成服务。CDM围绕大数据迁移上云和智能数据湖解决方案，提供了简单易用的迁移能力和多种数据源到数据湖的集成能力，降低了客户数据源迁移和集成的复杂性，有效的提高您数据迁移和集成的效率。更多详情请参见[云数据迁移服务](#)。

CDM进行数据迁移时，目标端为数据湖或其他大数据系统；源端可以是数据库也可以是对象存储。

#### CDM与DRS的区别：

- 目的端是大数据系统时，推荐使用CDM。
- 目的端是OLTP数据库或DWS时，推荐使用DRS迁移。

#### CDM与OMS的区别：

- OMS用于入云迁移，支持以下源端云服务商：亚马逊云、阿里云、微软云、百度云、青云、七牛云、腾讯云。
- CDM主要用于OBS数据迁移到数据湖或其他大数据系统，以便对数据进行开发、清洗、治理等。同时，整桶迁移建议使用OMS。

## 什么是对象存储迁移服务(OMS)?

对象存储迁移服务 ( Object Storage Migration Service, 简称OMS ) 是一种线上数据迁移服务, 帮助您将其他云服务商对象存储服务中的数据在线迁移至华为云的对象存储服务 ( Object Storage Service, OBS ) 中。简言之, 入云迁移、对象存储迁移。更多详情请参见[对象存储迁移服务](#)。

**OMS主要功能有以下两个:**

- 线上数据迁移服务: 帮助用户把对象存储数据从其他云服务商的公有云轻松、平滑地迁移上云。
- 跨区域的复制: 指的是华为云各个Region之间的数据复制和备份。

目前支持以下他云对象存储数据的入云迁移: 亚马逊云、阿里云、微软云、百度云、华为云、金山云、青云、七牛云、腾讯云。

**云数据迁移CDM服务也同样支持对象存储数据迁移, 两者的区别为:**

- OMS用于他云到华为云的数据迁移。
- CDM主要用于OBS数据迁移到数据湖或其他大数据系统, 以便对数据进行开发、清洗、治理等。

## 什么是数据复制服务(DRS)?

数据复制服务 ( Data Replication Service, 简称DRS ) 是一种易用、稳定、高效、用于数据库实时迁移和数据库实时同步的云服务。DRS适合迁移OLTP->OLTP、OLTP->DWS的场景都可以由DRS来完成数据迁移。即主流数据库到数据库 ( 含第三方数据库 ) 的场景, 使用DRS进行迁移。更多详情请参见[数据复制服务](#)。

**目前支持的数据库链路有:**

自建/他云MySQL->RDS for MySQL

自建/他云PostgreSQL->RDS for PostgreSQL

自建/他云MongoDB->DDS

Oracle->RDS for MySQL

.....

**DRS与CDM的区别:**

- DRS的目的端为数据库系统, 例如MySQL、MongoDB等。
- CDM的目的端主要为数据湖或其他大数据系统, 例如MRS HDFS、FusionInsight HDFS。

**DRS和UGO的区别:**

- DRS是针对数据的全量/增量迁移或数据同步。
- UGO用于异构数据库迁移前的评估、结构迁移和语法转化。

## 什么是主机迁移服务(SMS)?

主机迁移服务 ( Server Migration Service, 简称SMS ) 是一种P2V/V2V迁移服务, 可以帮您把X86物理服务器或者私有云、公有云平台上的虚拟机迁移到华为云弹性云服务器上, 从而帮助您轻松地把服务器上的应用和数据迁移到华为云。更多详情请参见[主机迁移服务](#)。



主机迁移服务 SMS 是一种P2V/V2V迁移服务，可以把X86物理服务器、私有云或公有云平台上的虚拟机迁移到华为ECS上。

## 什么是数据库和应用迁移(UGO)?

数据库和应用迁移 UGO ( Database and Application Migration UGO, 简称UGO ) 是专注于异构数据库结构迁移的专业服务。可将数据库中的DDL、业务程序中封装的数据库SQL一键自动将语法转换为华为云GaussDB/RDS的SQL语法，通过预迁移评估、结构迁移两大核心功能和自动化语法转换，提前识别可能存在的改造工作、提高转化率、最大化降低用户数据库迁移成本。更多详情请参见[数据库和应用迁移](#)。

简言之，UGO用于异构数据库迁移前的数据库评估、结构迁移、语法转化。

## 什么是数据快递服务(DES)?

数据快递服务 ( Data Express Service, 简称DES ) 是一种海量数据传输解决方案，支持TB到PB级数据上云，通过Teleport设备或硬盘（外置USB接口、SATA接口、SAS接口类型）向华为云传输大量数据，致力于解决海量数据传输网络成本高、传输时间长等难题。更多详情请参见[数据快递服务](#)。

## 各个数据迁移服务区别

表 1-1 各个数据迁移服务区别

| 服务名          | 主要功能   | 与其他服务的区别   |
|--------------|--|--|
| 云数据迁移 CDM    | <ul style="list-style-type: none"><li>大数据迁移上云</li><li>多种数据源到数据湖的迁移</li></ul>                   | <b>与DRS的区别：</b><br>数据库迁移使用DRS；到大数据系统的迁移使用CDM。  |
| 对象存储迁移服务 OMS | 对象存储迁移 <ul style="list-style-type: none"><li>他云对象存储数据迁移到华为云</li><li>华为云各Region间的数据迁移</li></ul> | <b>与CDM的区别：</b><br>OMS用于他云到华为云的数据迁移；CDM主要用于OBS数据迁移到数据湖或其他大数据系统，以便对数据进行开发、清洗、治理等。   |
| 数据复制服务 DRS   | 支持主流数据库到华为云的入云和出云迁移 <ul style="list-style-type: none"><li>数据库在线迁移</li><li>数据库实时同步</li></ul>    | <ul style="list-style-type: none"><li><b>与CDM的区别：</b><br/>数据库迁移使用DRS；到大数据系统的迁移使用CDM。</li><li><b>与UGO的区别：</b><br/>DRS支持同构和异构的数据库迁移/同步；UGO用于异构数据库的结构迁移、数据库迁移前评估、语法迁移等。</li></ul> |
| 主机迁移服务 SMS   | 主机迁移<br>含物理机到华为云、其他自建或他云虚拟机到华为云  | -  |

| 服务名          | 主要功能  | 与其他服务的区别   |
|--------------|---|--|
| 数据库和应用迁移 UGO | <ul style="list-style-type: none"><li>数据库结构迁移</li><li>数据库迁移前评估</li><li>语法迁移</li></ul> | <b>与DRS的区别：</b><br>DRS支持同构和异构的数据库迁移/同步；UGO用于异构数据库的结构迁移、数据库迁移前评估、语法迁移等。 |
| 数据快递服务 DES   | <ul style="list-style-type: none"><li>海量数据，支持TB级到PB级数据上云</li><li>使用物理介质</li></ul>     | -  |

## 1.2 CDM 有哪些优势？

云数据迁移（Cloud Data Migration，简称CDM）服务基于分布式计算框架，利用并行化处理技术，使用CDM迁移数据的优势如表1-2所示。

表 1-2 CDM 优势

| 优势项  | 用户自行开发   | CDM   |
|------|--|---|
| 易使用  | 自行准备服务器资源，安装配置必要的软件并进行配置，等待时间长。<br>程序在读写两端会根据数据源类型，使用不同的访问接口，一般是数据源提供的对外接口，例如JDBC、原生API等，因此在开发脚本时需要依赖大量的库、SDK等，开发管理成本较高。 | CDM提供了Web化的管理控制台，通过Web页实时开通服务。<br>用户只需要通过可视化界面对数据源和迁移任务进行配置，服务会对数据源和任务进行全面的管理和维护，用户只需关注数据迁移的具体逻辑，而不用关心环境等问题，极大降低了开发维护成本。<br>CDM还提供了REST API，支持第三方系统调用和集成。 |
| 实时监控 | 需要自行选型开发。  | 您可以使用云监控服务监控您的CDM集群，执行自动实时监控、告警和通知操作，帮助您更好地了解CDM集群的各项性能指标。  |
| 免运维  | 需要自行开发完善运维功能，自行保证系统可用性，尤其是告警及通知功能，否则只能人工值守。  | 使用CDM服务，用户不需要维护服务器、虚拟机等资源。CDM的日志，监控和告警功能，有异常可以及时通知相关人员，避免7*24小时人工值守。  |
| 高效率  | 在迁移过程中，数据读写过程都是由一个单一任务完成的，受限于资源，整体性能较低，对于海量数据场景通常不能满足要求。   | CDM任务基于分布式计算框架，自动将任务切分为独立的子任务并行执行，能够极大提高数据迁移的效率。针对Hive、HBase、MySQL、DWS（数据仓库服务）数据源，使用高效的数据导入接口导入数据。  |

| 优势项      | 用户自行开发   | CDM  |
|----------|--|--|
| 多种数据源支持  | 数据源类型繁杂，针对不同数据源开发不同的任务，脚本数量成千上万。   | 支持数据库、Hadoop、NoSQL、数据仓库、文件等多种类型的数据源。   |
| 多种网络环境支持 | 随着云计算技术的发展，用户数据可能存在于各种环境中，例如公有云、自建/托管IDC、混合场景等。在异构环境中进行数据迁移需要考虑网络连通性等因素，给开发和维护都带来较大难度。 | 无论数据是在用户本地自建的IDC中（Internet Data Center，互联网数据中心）、云服务中、第三方云中，或者使用ECS自建的数据库或文件系统中，CDM均可帮助用户轻松应对各种数据迁移场景，包括数据上云，云上数据交换，以及云上数据回流本地业务系统。 |

## 1.3 CDM 有哪些安全防护？

CDM是一个完全托管的服务，提供了以下安全防护能力保护用户数据安全。

- 实例隔离：CDM服务的用户只能使用自己创建的实例，实例和实例之间是相互隔离的，不可相互访问。
- 系统加固：CDM实例的操作系统进行了特别的安全加固，攻击者无法从Internet访问CDM实例的操作系统。
- 密钥加密：用户在CDM上创建连接输入的各种数据源的密钥，CDM均采用高强度加密算法保存在CDM数据库。
- 无中间存储：数据在迁移的过程中，CDM只处理数据映射和转换，而不会存储任何用户数据或片段。

## 1.4 如何降低 CDM 使用成本？

如果是迁移公网的数据上云，可以使用NAT网关服务，实现CDM服务与子网中的其他弹性云服务器共享弹性IP，可以更经济、更方便的通过Internet迁移本地数据中心或第三方云上的数据。

具体操作如下：

1. 假设已经创建好了CDM集群（无需为CDM集群绑定专用弹性IP），记录下CDM集群所在的VPC和子网。
2. 创建NAT网关，注意选择和CDM集群相同的VPC、子网。
3. 创建完NAT网关后，回到NAT网关控制台列表，单击创建好的网关名称，然后选择“添加SNAT规则”。

图 1-1 添加 SNAT 规则

### 添加SNAT规则

\* 子网名称:  ↻ ?  
192.168.2.0/24

\* 弹性IP:  ↻ ?

确定 取消

4. 选择子网和弹性IP，如果没有弹性IP，需要先申请一个。

完成之后，就可以到CDM控制台，通过Internet迁移公网的数据上云了。例如：迁移本地数据中心FTP服务器上的文件到OBS、迁移第三方云上关系型数据库到云服务RDS。

## 1.5 CDM 未使用数据传输功能时，是否会计费？

CDM集群运行状态下，即便未使用也是正常计费的。

如果长期不使用建议删除集群，需要的时候再创建集群。CDM集群计费详情请参考[价格详情](#)。

## 1.6 已购买包年包月的 CDM 套餐包，为什么还会产生按需计费的费用？

请您先确认套餐包和实际的CDM集群是否具有相同区域和规格，如果非相同区域和规格，则无法使用套餐包。CDM集群规格和区域可以通过进入CDM主界面，进入“集群管理”，单击集群列表中的集群名称查看。

如果套餐包和实际的CDM集群具有相同区域和规格，则以下情况也会产生按需费用：

如果您先购买按需计费增量包，再购买套餐包，则在购买套餐包之前已经产生的费用以按需计费结算，购买套餐包之后的费用按套餐包计时。

## 1.7 如何查看套餐包的剩余时长？

套餐包的剩余时长需要在官网查询。

您可以进入华为云官网，在用户名下拉列表中选择“费用中心”，然后进入“订单管理 - 续费管理”查看对应套餐包的剩余时长。

## 1.8 套餐包到期未续订或按需资源欠费时，我的数据会保留吗？

云服务进入宽限期/保留期后，华为云将会通过邮件、短信等方式向您发送提醒，提醒您续订或充值。**保留期到期仍未续订或充值，存储在云服务中的数据将被删除、云服务资源将被释放。**

- 宽限期：指客户的包周期资源到期未续订或按需资源欠费时，华为云提供给客户进行续费与充值的时间，宽限期内客户可正常访问及使用云服务。
- 保留期：指宽限期到期后客户的包周期资源仍未续订或按需资源仍未缴清欠款，将进入保留期。保留期内客户不能访问及使用云服务，但对客户存储在云服务中的数据仍予以保留。

华为云宽限期和保留期时长设定请参考[宽限期保留期](#)。

## 1.9 CDM 可以跨账户使用吗？

CDM不支持跨账户使用，可以通过授权给同一账户IAM子用户使用。

IAM用户授权操作步骤如下：

### 1. [创建用户组并授权](#)

在IAM控制台创建用户组，并授予CDM集群只读权限“CDM ReadOnlyAccess”。

### 2. [创建用户并加入用户组](#)

在IAM控制台创建用户，并将其加入1中创建的用户组。

### 3. [用户登录](#)并验证权限

新建的用户登录控制台，切换至授权区域，验证权限：

- 在“服务列表”中选择“云数据迁移服务”，进入CDM主界面查看集群，若未提示权限不足，表示“CDM ReadOnlyAccess”已生效。
- 在“服务列表”中选择除CDM服务外的任一服务，若提示权限不足，表示“CDM ReadOnlyAccess”已生效。

## 1.10 CDM 集群是否支持升级操作？

CDM集群目前不支持升级操作，如果需要使用高版本集群则需要重新创建。

## 1.11 CDM 迁移性能如何？

单个cdm.large规格实例理论上可以支持1TB~8TB/天的数据迁移，实际传输速率受公网带宽、集群规格、文件读写速度、作业并发数设置、磁盘读写性能等因素影响。更多详情请参见[性能白皮书](#)。

## 1.12 CDM 不同集群规格对应并发的作业数是多少？

CDM通过数据迁移作业，将源端数据迁移到目的端数据源中。其中，主要运行逻辑如下：

1. 数据迁移作业提交运行后，CDM会根据作业配置中的“抽取并发数”参数，将每个作业拆分为多个Task，即作业分片。

### 📖 说明

- 不同源端数据源的作业分片维度有所不同，因此某些作业可能出现未严格按作业“抽取并发数”参数分片的情况。
2. CDM依次将Task提交给运行池运行。根据集群配置管理中的“最大抽取并发数”参数，超出规格的Task排队等待运行。

### 如何调整抽取并发数

1. 集群最大抽取并发数的设置与CDM集群规格有关，并发数上限建议配置为vCPU核数\*2，如表1-3所示。

表 1-3 集群最大抽取并发数配置建议

| 规格名称        | vCPUs/内存  | 集群并发数上限参考 |
|-------------|-----------|-----------|
| cdm.large   | 8核 16GB   | 16        |
| cdm.xlarge  | 16核 32GB  | 32        |
| cdm.4xlarge | 64核 128GB | 128       |

图 1-2 集群最大抽取并发数配置

The screenshot shows the 'Configuration Management' (配置管理) tab in the CDM interface. The 'Maximum Concurrent Extraction' (最大抽取并发数) parameter is highlighted with a red box and set to the value 16. Other visible settings include 'Scheduled Backup' (定时备份) which is turned off, and an 'Environment Variables' (环境变量) text area which is currently empty. A 'Save' (保存) button is located at the bottom of the configuration panel.

2. 作业抽取并发数的配置原则如下：
  - a. 迁移的目的端为文件时，CDM不支持多并发，此时应配置为单进程抽取数据。
  - b. 表中每行数据大小为1MB以下的可以设置多并发抽取，超过1MB的建议单线程抽取数据。
  - c. 作业抽取并发数可参考集群最大抽取并发数配置，但不建议超过集群最大抽取并发数上限。
  - d. 目的端为DLI数据源时，抽取并发数建议配置为1，否则可能会导致写入失败。

**图 1-3** 作业抽取并发数配置

任务配置

作业失败重试 ?

作业分组 ?  添加 编辑 删除

是否定时执行

隐藏高级属性

抽取并发数 ?

分片重试次数 ?

是否写入脏数据 ?

开启限速 ?

## 1.13 CDM 集群可以关机吗？

2.9.1.200版本以后的集群已不支持集群定时关机、自动关机功能。

# 2 功能类

## 2.1 是否支持增量迁移？

### 问题描述

CDM是否支持增量迁移？

### 解决方案

CDM支持增量数据迁移。

利用定时任务配置和时间宏变量函数等参数，可支持以下场景的增量数据迁移：

- 文件增量迁移
- 关系数据库增量迁移
- HBase/CloudTable增量迁移

详情请参见[增量迁移](#)。

## 2.2 是否支持字段转换？

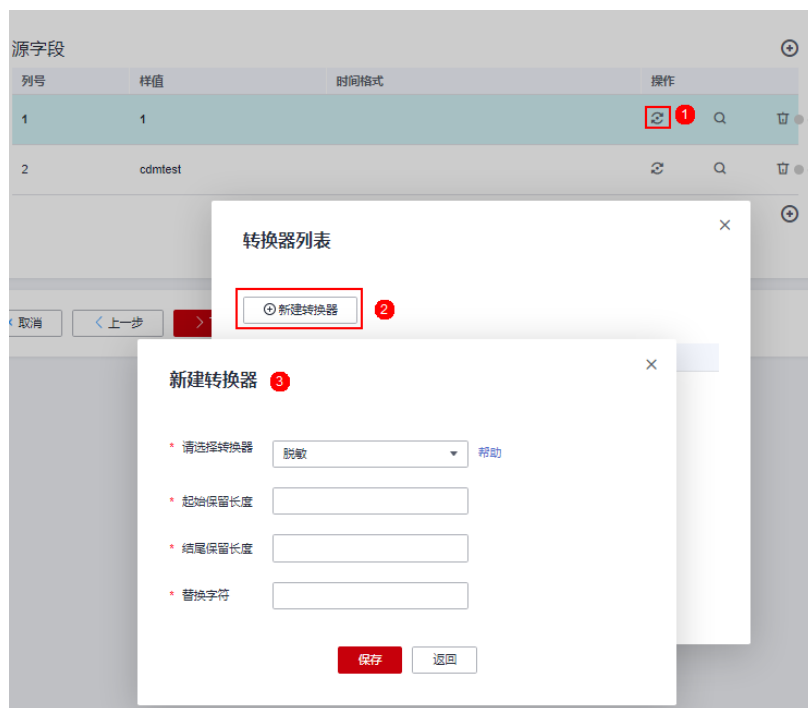
支持，CDM支持以下字段转换器：

- [脱敏](#)
- [去前后空格](#)
- [字符串反转](#)
- [字符串替换](#)
- [表达式转换](#)

在创建表/文件迁移作业的字段的映射界面，可新建字段转换器，如下图所示。



图 2-1 新建字段转换器



## 脱敏

隐藏字符串中的关键信息，例如要将“12345678910”转换为“123\*\*\*\*8910”，则配置如下：

- “起始保留长度”为“3”。
- “结尾保留长度”为“4”。
- “替换字符”为“\*”。

## 去前后空格

自动去字符串前后的空值，不需要配置参数。

## 字符串反转

自动反转字符串，例如将“ABC”转换为“CBA”，不需要配置参数。

## 字符串替换

替换字符串，需要用户配置被替换的对象，以及替换后的值。

## 表达式转换

使用JSP表达式语言（Expression Language）对当前字段或整行数据进行转换。JSP表达式语言可以用来创建算术和逻辑表达式。在表达式内可以使用整型数，浮点数，字符串，常量true、false和null。

数据进行转换过程中，替换内容包含特殊字符时，需要先使用\将该字符转义成普通字符。

- 表达式支持以下两个环境变量：
  - value: 当前字段值。
  - row: 当前行, 数组类型。
- 表达式支持的工具类用法罗列如下, 未列出即表示不支持:
  - a. 如果当前字段为字符串类型, 将字符串全部转换为小写, 例如将“aBC”转换为“abc”。  
表达式: `StringUtils.toLowerCase(value)`
  - b. 将当前字段的字符串全部转为大写。  
表达式: `StringUtils.toUpperCase(value)`
  - c. 如果想将第1个日期字段格式从“2018-01-05 15:15:05”转换为“20180105”。  
表达式: `DateUtils.format(DateUtils.parseDate(row[0], "yyyy-MM-dd HH:mm:ss"), "yyyyMMdd")`
  - d. 如果想将时间戳转换成“yyyy-MM-dd hh:mm:ss”格式的日期字符串的类型, 例如字段值为“1701312046588”, 转换后为“2023-11-30 10:40:46”。  
表达式: `DateUtils.format(NumberUtils.toLong(value), "yyyy-MM-dd HH:mm:ss")`
  - e. 如果想将“yyyy-MM-dd hh:mm:ss”格式的日期字符串转换成时间戳的类型。  
表达式: `DateUtils.getTime(DateUtils.parseDate(value, "yyyy-MM-dd hh:mm:ss"))`
  - f. 如果当前字段值为“yyyy-MM-dd”格式的日期字符串, 需要截取年, 例如字段值为“2017-12-01”, 转换后为“2017”。  
表达式: `StringUtils.substringBefore(value, "-")`
  - g. 如果当前字段值为数值类型, 转换后值为当前值的两倍。  
表达式: `value*2`
  - h. 如果当前字段值为“true”, 转换后为“Y”, 其它值则转换后为“N”。  
表达式: `value=="true"? "Y": "N"`
  - i. 如果当前字段值为字符串类型, 当为空时, 转换为“Default”, 否则不转换。  
表达式: `empty value? "Default":value`
  - j. 如果想将日期字段格式从“2018/01/05 15:15:05”转换为“2018-01-05 15:15:05”。  
表达式: `DateUtils.format(DateUtils.parseDate(value, "yyyy/MM/dd HH:mm:ss"), "yyyy-MM-dd HH:mm:ss")`
  - k. 获取一个36位的UUID (Universally Unique Identifier, 通用唯一识别码)。  
表达式: `CommonUtils.randomUUID()`
  - l. 如果当前字段值为字符串类型, 将首字母转换为大写, 例如将“cat”转换为“Cat”。  
表达式: `StringUtils.capitalize(value)`
  - m. 如果当前字段值为字符串类型, 将首字母转换为小写, 例如将“Cat”转换为“cat”。  
表达式: `StringUtils.uncapitalize(value)`

- n. 如果当前字段值为字符串类型，使用空格填充为指定长度，并且将字符串居中，当字符串长度不小于指定长度时不转换，例如将“ab”转换为长度为4的“ab”。
- 表达式：`StringUtils.center(value,4)`
- o. 删除字符串末尾的一个换行符（包括“\n”、“\r”或者“\r\n”），例如将“abc\r\n\r\n”转换为“abc\r\n”。
- 表达式：`StringUtils.chomp(value)`
- p. 如果字符串中包含指定的字符串，则返回布尔值true，否则返回false。例如“abc”中包含“a”，则返回true。
- 表达式：`StringUtils.contains(value,"a")`
- q. 如果字符串中包含指定字符串的任一字符，则返回布尔值true，否则返回false。例如“zzabyycdxx”中包含“z”或“a”任意一个，则返回true。
- 表达式：`StringUtils.containsAny(value,"za")`
- r. 如果字符串中不包含指定的所有字符，则返回布尔值true，包含任意一个字符则返回false。例如“abz”中包含“xyz”里的任意一个字符，则返回false。
- 表达式：`StringUtils.containsNone(value,"xyz")`
- s. 如果当前字符串只包含指定字符串中的字符，则返回布尔值true，包含任意一个其它字符则返回false。例如“abab”只包含“abc”中的字符，则返回true。
- 表达式：`StringUtils.containsOnly(value,"abc")`
- t. 如果字符串为空或null，则转换为指定的字符串，否则不转换。例如将空字符转换为null。
- 表达式：`StringUtils.defaultIfEmpty(value,null)`
- u. 如果字符串以指定的后缀结尾（包括大小写），则返回布尔值true，否则返回false。例如“abcdef”后缀不为null，则返回false。
- 表达式：`StringUtils.endsWith(value,null)`
- v. 如果字符串和指定的字符串完全一样（包括大小写），则返回布尔值true，否则返回false。例如比较字符串“abc”和“ABC”，则返回false。
- 表达式：`StringUtils.equals(value,"ABC")`
- w. 从字符串中获取指定字符串的第一个索引，没有则返回整数-1。例如从“aabaabaa”中获取“ab”的第一个索引1。
- 表达式：`StringUtils.indexOf(value,"ab")`
- x. 从字符串中获取指定字符串的最后一个索引，没有则返回整数-1。例如从“aFkyk”中获取“k”的最后一个索引4。
- 表达式：`StringUtils.lastIndexOf(value,"k")`
- y. 从字符串中指定的位置往后查找，获取指定字符串的第一个索引，没有则转换为“-1”。例如“aabaabaa”中索引3的后面，第一个“b”的索引是5。
- 表达式：`StringUtils.indexOf(value,"b",3)`
- z. 从字符串获取指定字符串中任一字符的第一个索引，没有则返回整数-1。例如从“zzabyycdxx”中获取“z”或“a”的第一个索引0。
- 表达式：`StringUtils.indexOfAny(value,"za")`
- aa. 如果字符串仅包含Unicode字符，返回布尔值true，否则返回false。例如“ab2c”中包含非Unicode字符，返回false。
- 表达式：`StringUtils.isAlpha(value)`

- ab. 如果字符串仅包含Unicode字符或数字，返回布尔值true，否则返回false。例如“ab2c”中仅包含Unicode字符和数字，返回true。  
表达式: `StringUtils.isAlphanumeric(value)`
- ac. 如果字符串仅包含Unicode字符、数字或空格，返回布尔值true，否则返回false。例如“ab2c”中仅包含Unicode字符和数字，返回true。  
表达式: `StringUtils.isAlphanumericSpace(value)`
- ad. 如果字符串仅包含Unicode字符或空格，返回布尔值true，否则返回false。例如“ab2c”中包含Unicode字符和数字，返回false。  
表达式: `StringUtils.isAlphaSpace(value)`
- ae. 如果字符串仅包含ASCII可打印字符，返回布尔值true，否则返回false。例如“!ab-c~”返回true。  
表达式: `StringUtils.isAsciiPrintable(value)`
- af. 如果字符串为空或null，返回布尔值true，否则返回false。  
表达式: `StringUtils.isEmpty(value)`
- ag. 如果字符串中仅包含Unicode数字，返回布尔值true，否则返回false。  
表达式: `StringUtils.isNumeric(value)`
- ah. 获取字符串最左端的指定长度的字符，例如获取“abc”最左端的2位字符“ab”。  
表达式: `StringUtils.left(value,2)`
- ai. 获取字符串最右端的指定长度的字符，例如获取“abc”最右端的2位字符“bc”。  
表达式: `StringUtils.right(value,2)`
- aj. 将指定字符串拼接至当前字符串的左侧，需同时指定拼接后的字符串长度，如果当前字符串长度不小于指定长度，则不转换。例如将“yz”拼接到“bat”左侧，拼接后长度为8，则转换后为“zyzybat”。  
表达式: `StringUtils.leftPad(value,8,"yz")`
- ak. 将指定字符串拼接至当前字符串的右侧，需同时指定拼接后的字符串长度，如果当前字符串长度不小于指定长度，则不转换。例如将“yz”拼接到“bat”右侧，拼接后长度为8，则转换后为“batzyzy”。  
表达式: `StringUtils.rightPad(value,8,"yz")`
- al. 如果当前字段为字符串类型，获取当前字符串的长度，如果该字符串为null，则返回0。  
表达式: `StringUtils.length(value)`
- am. 如果当前字段为字符串类型，删除其中所有的指定字符串，例如从“queued”中删除“ue”，转换后为“qd”。  
表达式: `StringUtils.remove(value,"ue")`
- an. 如果当前字段为字符串类型，移除当前字段末尾指定的子字符串。指定的子字符串若不在当前字段的末尾，则不转换，例如移除当前字段“www.domain.com”后的“.com”。  
表达式: `StringUtils.removeEnd(value,".com")`
- ao. 如果当前字段为字符串类型，移除当前字段开头指定的子字符串。指定的子字符串若不在当前字段的开头，则不转换，例如移除当前字段“www.domain.com”前的“www.”。  
表达式: `StringUtils.removeStart(value,"www.")`

- ap. 如果当前字段为字符串类型，替换当前字段中所有的指定字符串，例如将“aba”中的“a”用“z”替换，转换后为“zbz”。
- 表达式：`StringUtils.replace(value,"a","z")`
- 替换内容包含特殊字符时，需要先把该字符转义成普通字符，例如，客户想通过该表达式把字符串中 \t 去掉时，需要配置为：`StringUtils.replace(value,"\\t","")`（即把 \ 再次转义）。
- aq. 如果当前字段为字符串类型，一次替换字符串中的多个字符，例如将字符串“hello”中的“h”用“j”替换，“o”用“y”替换，转换后为“jelly”。
- 表达式：`StringUtils.replaceChars(value,"ho","jy")`
- ar. 如果字符串以指定的前缀开头（区分大小写），则返回布尔值true，否则返回false，例如当前字符串“abcdef”以“abc”开头，则返回true。
- 表达式：`StringUtils.startsWith(value,"abc")`
- as. 如果当前字段为字符串类型，去除字段中首、尾处所有指定的字符，例如去除“abcyx”中首尾所有的“x”、“y”、“z”和“b”，转换后为“abc”。
- 表达式：`StringUtils.strip(value,"xyzb")`
- at. 如果当前字段为字符串类型，去除字段末尾所有指定的字符，例如去除当前字段末尾的“abc”字符串。
- 表达式：`StringUtils.stripEnd(value,"abc")`
- au. 如果当前字段为字符串类型，去除字段开头所有指定的字符，例如去除当前字段开头的空格。
- 表达式：`StringUtils.stripStart(value,null)`
- av. 如果当前字段为字符串类型，获取字符串指定位置后（索引从0开始，包括指定位置的字符）的子字符串，指定位置如果为负数，则从末尾往前计算位置，末尾第一位为-1。例如获取“abcde”索引为2的字符（即c）及之后的字符串，则转换后为“cde”。
- 表达式：`StringUtils.substring(value,2)`
- aw. 如果当前字段为字符串类型，获取字符串指定区间（索引从0开始，区间起点包括指定位置的字符，区间终点不包含指定位置的字符）的子字符串，区间位置如果为负数，则从末尾往前计算位置，末尾第一位为-1。例如获取“abcde”第2个字符（即c）及之后、第4个字符（即e）之前的字符串，则转换后为“cd”。
- 表达式：`StringUtils.substring(value,2,4)`
- ax. 如果当前字段为字符串类型，获取当前字段里第一个指定字符后的子字符串。例如获取“abcba”中第一个“b”之后的子字符串，转换后为“cba”。
- 表达式：`StringUtils.substringAfter(value,"b")`
- ay. 如果当前字段为字符串类型，获取当前字段里最后一个指定字符后的子字符串。例如获取“abcba”中最后一个“b”之后的子字符串，转换后为“a”。
- 表达式：`StringUtils.substringAfterLast(value,"b")`
- az. 如果当前字段为字符串类型，获取当前字段里第一个指定字符前的子字符串。例如获取“abcba”中第一个“b”之前的子字符串，转换后为“a”。
- 表达式：`StringUtils.substringBefore(value,"b")`
- ba. 如果当前字段为字符串类型，获取当前字段里最后一个指定字符前的子字符串。例如获取“abcba”中最后一个“b”之前的子字符串，转换后为“abc”。

- 表达式: `StringUtils.substringBeforeLast(value,"b")`
- bb. 如果当前字段为字符串类型, 获取嵌套在指定字符串之间的子字符串, 没有匹配的则返回null。例如获取“tagabctag”中“tag”之间的子字符串, 转换后为“abc”。
- 表达式: `StringUtils.substringBetween(value,"tag")`
- bc. 如果当前字段为字符串类型, 删除当前字符串两端的控制字符 (`char≤32`), 例如删除字符串前后的空格。
- 表达式: `StringUtils.trim(value)`
- bd. 将当前字符串转换为字节, 如果转换失败, 则返回0。
- 表达式: `NumberUtils.toByte(value)`
- be. 将当前字符串转换为字节, 如果转换失败, 则返回指定值, 例如指定值配置为1。
- 表达式: `NumberUtils.toByte(value, 1)`
- bf. 将当前字符串转换为Double数值, 如果转换失败, 则返回0.0d。
- 表达式: `NumberUtils.toDouble(value)`
- bg. 将当前字符串转换为Double数值, 如果转换失败, 则返回指定值, 例如指定值配置为1.1d。
- 表达式: `NumberUtils.toDouble(value, 1.1d)`
- bh. 将当前字符串转换为Float数值, 如果转换失败, 则返回0.0f。
- 表达式: `NumberUtils.toFloat(value)`
- bi. 将当前字符串转换为Float数值, 如果转换失败, 则返回指定值, 例如配置指定值为1.1f。
- 表达式: `NumberUtils.toFloat(value, 1.1f)`
- bj. 将当前字符串转换为Int数值, 如果转换失败, 则返回0。
- 表达式: `NumberUtils.toInt(value)`
- bk. 将当前字符串转换为Int数值, 如果转换失败, 则返回指定值, 例如配置指定值为1。
- 表达式: `NumberUtils.toInt(value, 1)`
- bl. 将字符串转换为Long数值, 如果转换失败, 则返回0。
- 表达式: `NumberUtils.toLong(value)`
- bm. 将当前字符串转换为Long数值, 如果转换失败, 则返回指定值, 例如配置指定值为1L。
- 表达式: `NumberUtils.toLong(value, 1L)`
- bn. 将字符串转换为Short数值, 如果转换失败, 则返回0。
- 表达式: `NumberUtils.toShort(value)`
- bo. 将当前字符串转换为Short数值, 如果转换失败, 则返回指定值, 例如配置指定值为1。
- 表达式: `NumberUtils.toShort(value, 1)`
- bp. 将当前IP字符串转换为Long数值, 例如将“10.78.124.0”转换为Long数值是“172915712”。
- 表达式: `CommonUtils.ipToLong(value)`
- bq. 从网络读取一个IP与物理地址映射文件, 并存放到Map集合, 这里的URL是IP与地址映射文件存放地址, 例如“`http://10.114.205.45:21203/sqoop/IpList.csv`”。

- 表达式: `HttpsUtils.downloadMap("url")`
- br. 将IP与地址映射对象缓存起来并指定一个key值用于检索, 例如“ipList”。
- 表达式: `CommonUtils.setCache("ipList", HttpsUtils.downloadMap("url"))`
- bs. 取出缓存的IP与地址映射对象。
- 表达式: `CommonUtils.getCache("ipList")`
- bt. 判断是否有IP与地址映射缓存。
- 表达式: `CommonUtils.cacheExists("ipList")`
- bu. 根据IP取出对应的详细地址: 国家\_省份\_城市\_运营商, 例如“1xx.78.124.0”对应的地址为“中国\_广东\_深圳\_电信”, 取不到对应地址则默认“\*\*\_\*\*\_\*\*\_\*\*”。如果需要, 可通过StringUtil类表达式对地址进行进一步拆分。
- 表达式:  
`CommonUtils.getMapValue(CommonUtils.ipToLong(value), CommonUtils.cacheExists("ipList"))?`  
`CommonUtils.getCache("ipList"):CommonUtils.setCache("ipList", HttpsUtils.downloadMap("url"))`
- bv. 根据指定的偏移类型 ( month/day/hour/minute/second ) 及偏移量 ( 正数表示增加, 负数表示减少 ), 将指定格式的时间转换为一个新时间, 例如将“2019-05-21 12:00:00”增加8个小时。
- 表达式: `DateUtils.getCurrentTimeByZone("yyyy-MM-dd HH:mm:ss", value, "hour", 8)`
- bw. 如果value值为空或者null时, 则返回字符串“aaa”, 否则返回value。
- 表达式: `StringUtils.defaultIfEmpty(value, "aaa")`

## 2.3 Hadoop 类型的数据源进行数据迁移时, 建议使用的组件版本有哪些?

建议使用的组件版本既可以作为目的端使用, 也可以作为源端使用。

表 2-1 建议使用的组件版本

| Hadoop类型                        | 组件    | 说明  |
|---------------------------------|-------|---|
| MRS/Apache/<br>FusionInsight HD | Hive  | 暂不支持2.x版本, 建议使用的版本:<br><ul style="list-style-type: none"> <li>• 1.2.X</li> <li>• 3.1.X</li> </ul> |
|                                 | HDFS  | 建议使用的版本:<br><ul style="list-style-type: none"> <li>• 2.8.X</li> <li>• 3.1.X</li> </ul>            |
|                                 | Hbase | 建议使用的版本:<br><ul style="list-style-type: none"> <li>• 2.1.X</li> <li>• 1.3.X</li> </ul>            |

## 2.4 数据源为 Hive 时支持哪些数据格式？

### 问题描述

云数据迁移服务，数据源为Hive时支持的数据格式有哪些？

### 解决方案

云数据迁移服务支持从Hive数据源读写的数据格式包括SequenceFile、TextFile、ORC、Parquet。

## 2.5 是否支持同步作业到其他集群？

### 问题描述

CDM是否支持同步作业到其他集群？

### 解决方案

CDM虽然不支持直接在不同集群间迁移作业，但是通过批量导出、批量导入作业的功能，可以间接实现集群间的作业迁移，方法如下：

1. 将CDM集群1中的所有作业批量导出，将作业的JSON文件保存到本地。  
由于安全原因，CDM导出作业时没有导出连接密码，连接密码全部使用“Add password here”替换。
2. 在本地编辑JSON文件，将“Add password here”替换为对应连接的正确密码。
3. 将编辑好的JSON文件批量导入到CDM集群2，实现集群1和集群2之间的作业同步。

## 2.6 是否支持批量创建作业？

### 问题描述

CDM是否支持批量创建作业？

### 解决方案

CDM可以通过批量导入的功能，实现批量创建作业，方法如下：

1. 手动创建一个作业。
2. 导出作业，将作业的JSON文件保存到本地。
3. 编辑JSON文件，参考该作业的配置，在JSON文件中批量复制出更多作业。
4. 将JSON文件导入CDM集群，实现批量创建作业。

您也可以参考[通过CDM算子批量创建分表迁移作业](#)，配合For Each算子，实现自动批量创建作业。



## 2.7 是否支持批量调度作业？

### 问题描述

CDM是否支持批量调度作业？

### 解决方案

支持。

1. 访问DataArts Studio服务的数据开发模块。
2. 在数据开发主界面的左侧导航栏，选择“数据开发 > 作业开发”，新建作业。
3. 拖动多个CDM Job节点至画布，然后再编排作业。

## 2.8 如何备份 CDM 作业？

### 问题描述

如何备份CDM作业？

### 解决方案

用户可以先通过CDM的批量导出功能，把所有作业脚本保存到本地，在需要的时候再重新创建集群、重新导入作业，实现作业备份。

## 2.9 如何解决 HANA 集群只有部分节点和 CDM 集群网络互通？

### 问题描述

HANA集群只有部分节点和CDM集群网络互通，怎么办？

### 解决方案

如果HANA集群只有部分节点和CDM网络互通，为确保CDM正常连接HANA集群，则需要进行如下配置：

1. 关闭HANA集群的Statement Routing开关。但须注意，关闭Statement Routing，会增加配置节点的压力。
2. 新建HANA连接时，在高级属性中添加属性“distribution”，并将值置为“off”。

完成配置后，CDM即可正常连接HANA集群。

## 2.10 如何使用 Java 调用 CDM 的 Rest API 创建数据迁移作业?

CDM提供了Rest API，可以通过程序调用实现自动化的作业创建或执行控制。

这里以CDM迁移MySQL数据库的表city1的数据到DWS的表city2为例，介绍如何使用Java调用CDM服务的REST API创建、启动、查询、删除该CDM作业。

需要提前准备以下数据：

1. 云账号的用户名、账号名和项目ID。
2. 创建一个CDM集群，并获取集群ID。  
获取方法：在集群管理界面，单击CDM集群名称可查看集群ID，例如“c110beff-0f11-4e75-8b10-da7cd882b0ef”。
3. 创建一个MySQL数据库和一个DWS数据库，并创建好表city1和表city2，创表语句如下：

```
MySQL:
create table city1(code varchar(10),name varchar(32));
insert into city1 values('NY','New York');
DWS:
create table city2(code varchar(10),name varchar(32));
```

4. 在CDM集群下，创建连接到MySQL的连接，例如连接名称为“mysqltestlink”。创建连接到DWS的连接，例如连接名称为“dwstestlink”。
5. 运行下述代码，依赖HttpClient包，建议使用4.5版本。Maven配置如下：

```
<project>
<modelVersion>4.0.0</modelVersion>
<groupId>cdm</groupId>
<artifactId>cdm-client</artifactId>
<version>1</version>
<dependencies>
<dependency>
<groupId>org.apache.httpcomponents</groupId>
<artifactId>httpclient</artifactId>
<version>4.5</version>
</dependency>
</dependencies>
</project>
```

### 代码示例

使用Java调用CDM服务的REST API创建、启动、查询、删除CDM作业的代码示例如下：

```
package cdmclient;
import java.io.IOException;
import org.apache.http.Header;
import org.apache.http.HttpEntity;
import org.apache.http.HttpHost;
import org.apache.http.auth.AuthScope;
import org.apache.http.auth.UsernamePasswordCredentials;
import org.apache.http.client.CredentialsProvider;
import org.apache.http.client.config.RequestConfig;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpDelete;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.methods.HttpPut;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.BasicCredentialsProvider;
```

```
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;
public class CdmClient {
    private final static String DOMAIN_NAME="云账号名";
    private final static String USER_NAME="云用户名";
    private final static String USER_PASSWORD="云用户密码";
    private final static String PROJECT_ID="项目ID";
    private final static String CLUSTER_ID="CDM集群ID";
    private final static String JOB_NAME="作业名称";
    private final static String FROM_LINKNAME="源连接名称";
    private final static String TO_LINKNAME="目的连接名称";
    private final static String IAM_ENDPOINT="IAM的Endpoint";
    private final static String CDM_ENDPOINT="CDM的Endpoint";
    private CloseableHttpClient httpClient;
    private String token;

    public CdmClient() {
        this.httpClient = createHttpClient();
        this.token = login();
    }

    private CloseableHttpClient createHttpClient() {
        CloseableHttpClient httpClient =HttpClients.createDefault();
        return httpClient;
    }

    private String login(){
        HttpPost httpPost = new HttpPost("https://"+IAM_ENDPOINT+"/v3/auth/tokens");
        String json =
            "{\r\n"+
            "\  \"auth\": {\r\n"+
            "\    \"identity\": {\r\n"+
            "\      \"methods\": [\r\n"+
            "\        \"password\": {\r\n"+
            "\          \"user\": {\r\n"+
            "\            \"name\": \""+USER_NAME+"\", \r\n"+
            "\            \"password\": \""+USER_PASSWORD+"\", \r\n"+
            "\            \"domain\": {\r\n"+
            "\              \"name\": \""+DOMAIN_NAME+"\" \r\n"+
            "\            }\r\n"+
            "\          }\r\n"+
            "\        }\r\n"+
            "\      }\r\n"+
            "\    }, \r\n"+
            "\    \"scope\": {\r\n"+
            "\      \"project\": {\r\n"+
            "\        \"name\": \"PROJECT_NAME\" \r\n"+
            "\      }\r\n"+
            "\    }\r\n"+
            "\  }\r\n"+
            "\}";
        try {
            StringEntity s = new StringEntity(json);
            s.setContentEncoding("UTF-8");
            s.setContentType("application/json");
            httpPost.setEntity(s);
            CloseableHttpResponse response = httpClient.execute(httpPost);
            Header tokenHeader = response.getFirstHeader("X-Subject-Token");
            String token = tokenHeader.getValue();
            System.out.println("Login successful");
            return token;
        } catch (Exception e) {
            throw new RuntimeException("login failed.", e);
        }
    }
    /*创建作业*/

    public void createJob(){
        HttpPost httpPost = new HttpPost("https://"+CDM_ENDPOINT+"/cdm/v1.0/"+PROJECT_ID+"/
```

```
clusters/"+CLUSTER_ID+"/cdm/job");
```

/\*\*此处JSON信息比较复杂，可以先在作业管理界面上创建一个作业，然后单击作业后的“作业JSON定义”，复制其中的JSON内容，格式化为Java字符串语法，然后粘贴到此处。

\*JSON消息体中一般只需要替换连接名、导入和导出的表名、导入导出表的字段列表、源表中用于分区的字段。  
\*\*/

```
String json =
"{\r\n"+
"\\"jobs\":[\r\n"+
"{\r\n"+
"\\"from-connector-name\":"generic-jdbc-connector",\r\n"+
"\\"name\":"+"JOB_NAME+",\r\n"+
"\\"to-connector-name\":"generic-jdbc-connector",\r\n"+
"\\"driver-config-values\":{\r\n"+
"\\"configs\":[\r\n"+
"{\r\n"+
"\\"inputs\":[\r\n"+
"{\r\n"+
"\\"name\":"throttlingConfig.numExtractors",\r\n"+
"\\"value\":"1",\r\n"+
"},\r\n"+
"\\"validators\":[],\r\n"+
"\\"type\":"JOB",\r\n"+
"\\"id\":30,\r\n"+
"\\"name\":"throttlingConfig",\r\n"+
"}\r\n"+
"]\r\n"+
"},\r\n"+
"\\"from-link-name\":"+"FROM_LINKNAME+",\r\n"+
"\\"from-config-values\":{\r\n"+
"\\"configs\":[\r\n"+
"{\r\n"+
"\\"inputs\":[\r\n"+
"{\r\n"+
"\\"name\":"fromJobConfig.schemaName",\r\n"+
"\\"value\":"sqoop",\r\n"+
"},\r\n"+
"},\r\n"+
"\\"name\":"fromJobConfig.tableName",\r\n"+
"\\"value\":"city1",\r\n"+
"},\r\n"+
"{\r\n"+
"\\"name\":"fromJobConfig.columnList",\r\n"+
"\\"value\":"code&name",\r\n"+
"},\r\n"+
"},\r\n"+
"\\"name\":"fromJobConfig.partitionColumn",\r\n"+
"\\"value\":"code",\r\n"+
"},\r\n"+
"]\r\n"+
"\\"validators\":[],\r\n"+
"\\"type\":"JOB",\r\n"+
"\\"id\":7,\r\n"+
"\\"name\":"fromJobConfig",\r\n"+
"}\r\n"+
"]\r\n"+
"},\r\n"+
"\\"to-link-name\":"+"TO_LINKNAME+",\r\n"+
"\\"to-config-values\":{\r\n"+
"\\"configs\":[\r\n"+
"{\r\n"+
"\\"inputs\":[\r\n"+
"{\r\n"+
"\\"name\":"toJobConfig.schemaName",\r\n"+
"\\"value\":"sqoop",\r\n"+
"},\r\n"+
"},\r\n"+
"}\r\n"+
```

```
"\name\": \"toJobConfig.tableName\", \r\n"+
"\value\": \"city2\" \r\n"+
"} \r\n"+
"{ \r\n"+
"\name\": \"toJobConfig.columnList\", \r\n"+
"\value\": \"code&name\" \r\n"+
"} \r\n"+
"{ \r\n"+
"\name\": \"toJobConfig.shouldClearTable\", \r\n"+
"\value\": \"true\" \r\n"+
"} \r\n"+
"} \r\n"+
"} \r\n"+
"\validators\": [], \r\n"+
"\type\": \"JOB\", \r\n"+
"\id\": 9, \r\n"+
"\name\": \"toJobConfig\" \r\n"+
"} \r\n"+
"} \r\n"+
"} \r\n"+
"} \r\n"+
"} \r\n"+
"} \r\n"+
}
try {
StringEntity s = new StringEntity(json);
s.setContentEncoding("UTF-8");
s.setContentType("application/json");
httpPost.setEntity(s);
httpPost.addHeader("X-Auth-Token", this.token);
httpPost.addHeader("X-Language", "en-us");
CloseableHttpResponse response = httpClient.execute(httpPost);
int status = response.getStatusLine().getStatusCode();
if(status == 200){
System.out.println("Create job successful.");
}else{
System.out.println("Create job failed.");
HttpEntity entity = response.getEntity();
System.out.println(EntityUtils.toString(entity));
}
} catch (Exception e) {
e.printStackTrace();
throw new RuntimeException("Create job failed.", e);
}
}
/*启动作业*/

public void startJob(){
HttpPut httpPut = new HttpPut("https://"+CDM_ENDPOINT+"/cdm/v1.0/"+PROJECT_ID+"/clusters/"+CLUSTER_ID+"/cdm/job/"+JOB_NAME+"/start");
String json = "";
try {
StringEntity s = new StringEntity(json);
s.setContentEncoding("UTF-8");
s.setContentType("application/json");
httpPut.setEntity(s);
httpPut.addHeader("X-Auth-Token", this.token);
httpPut.addHeader("X-Language", "en-us");
CloseableHttpResponse response = httpClient.execute(httpPut);
int status = response.getStatusLine().getStatusCode();
if(status == 200){
System.out.println("Start job successful.");
}else{
System.out.println("Start job failed.");
HttpEntity entity = response.getEntity();
System.out.println(EntityUtils.toString(entity));
}
} catch (Exception e) {
e.printStackTrace();
throw new RuntimeException("Start job failed.", e);
}
```

```
}  
}  
/*循环查询作业运行状态，直到作业运行结束。*/  
  
public void getJobStatus(){  
    HttpGet httpGet = new HttpGet("https://" + CDM_ENDPOINT + "/cdm/v1.0/" + PROJECT_ID + "/"  
    clusters/" + CLUSTER_ID + "/cdm/job/" + JOB_NAME + "/status");  
    try {  
        httpGet.addHeader("X-Auth-Token", this.token);  
        httpGet.addHeader("X-Language", "en-us");  
        boolean flag = true;  
        while(flag){  
            CloseableHttpResponse response = httpClient.execute(httpGet);  
            int status = response.getStatusLine().getStatusCode();  
            if(status == 200){  
                HttpEntity entity = response.getEntity();  
                String msg = EntityUtils.toString(entity);  
                if(msg.contains("\"status\": \"SUCCEEDED\"")){  
                    System.out.println("Job succeeded");  
                    break;  
                }else if (msg.contains("\"status\": \"FAILED\"")){  
                    System.out.println("Job failed.");  
                    break;  
                }else{  
                    Thread.sleep(1000);  
                }  
            }else{  
                System.out.println("Get job status failed.");  
                HttpEntity entity = response.getEntity();  
                System.out.println(EntityUtils.toString(entity));  
                break;  
            }  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
        throw new RuntimeException("Get job status failed.", e);  
    }  
}  
/*删除作业*/  
  
public void deleteJob(){  
    HttpDelete httpDelte = new HttpDelete("https://" + CDM_ENDPOINT + "/cdm/v1.0/" + PROJECT_ID + "/"  
    clusters/" + CLUSTER_ID + "/cdm/job/" + JOB_NAME);  
    try {  
        httpDelte.addHeader("X-Auth-Token", this.token);  
        httpDelte.addHeader("X-Language", "en-us");  
        CloseableHttpResponse response = httpClient.execute(httpDelte);  
        int status = response.getStatusLine().getStatusCode();  
        if(status == 200){  
            System.out.println("Delete job successful.");  
        }else{  
            System.out.println("Delete job failed.");  
            HttpEntity entity = response.getEntity();  
            System.out.println(EntityUtils.toString(entity));  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
        throw new RuntimeException("Delete job failed.", e);  
    }  
}  
/*关闭*/  
  
public void close(){  
    try {  
        httpClient.close();  
    } catch (IOException e) {  
        throw new RuntimeException("Close failed.", e);  
    }  
}
```

```
}  
public static void main(String[] args){  
    CdmClient cdmClient = new CdmClient();  
    cdmClient.createJob();  
    cdmClient.startJob();  
    cdmClient.getJobStatus();  
    cdmClient.deleteJob();  
    cdmClient.close();  
}  
}
```

## 2.11 如何将云下内网或第三方云上的私网与 CDM 连通？

很多企业会把关键数据源建设在内网，例如数据库、文件服务器等。由于CDM运行在云上，如果要通过CDM迁移内网数据到云上的话，可以通过以下几种方式连通内网和CDM的网络：

- 如果目标数据源为云下的数据库，则需要通过公网或者专线打通网络。通过公网互通时，需确保CDM集群已绑定EIP、CDM云上安全组出方向放通云下数据源所在的主机、数据源所在的主机可以访问公网且防火墙规则已开放连接端口。
- 在本地数据中心和云服务VPC之间建立VPN通道。
- 通过NAT（网络地址转换，Network Address Translation）或端口转发，以代理的方式访问。

这里重点介绍如何通过端口转发工具来实现访问内部数据，流程如下：

1. 找一台windows机器作为网关，该机器必须可以直接访问Internet，同时可以访问内网。
2. 在该机器上安装端口映射工具（IPOP）。
3. 通过端口映射工具（IPOP）配置端口映射。

---

### 须知

长时间将内网数据库暴露在公网会有安全风险，迁移数据完成后，请及时停止端口映射。

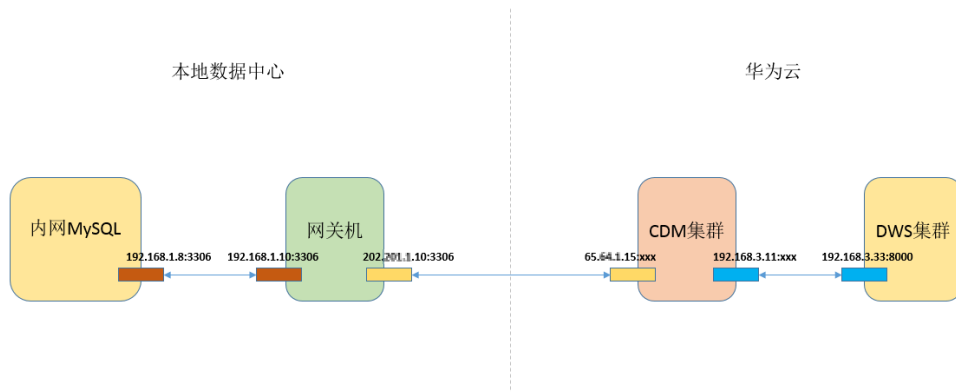
---

### 场景描述

这里假设是将内网MySQL迁移到云服务DWS，网络拓扑样例如图2-2所示。

图中的内网既可以是企业自己的数据中心，也可以是在第三方云的虚拟数据中心私网。

图 2-2 网络拓扑样例



## 操作步骤

**步骤1** 找一台Windows机器作为网关机，该机器同时配置内网和外网IP。通过以下测试来确保网关机的服务要求：

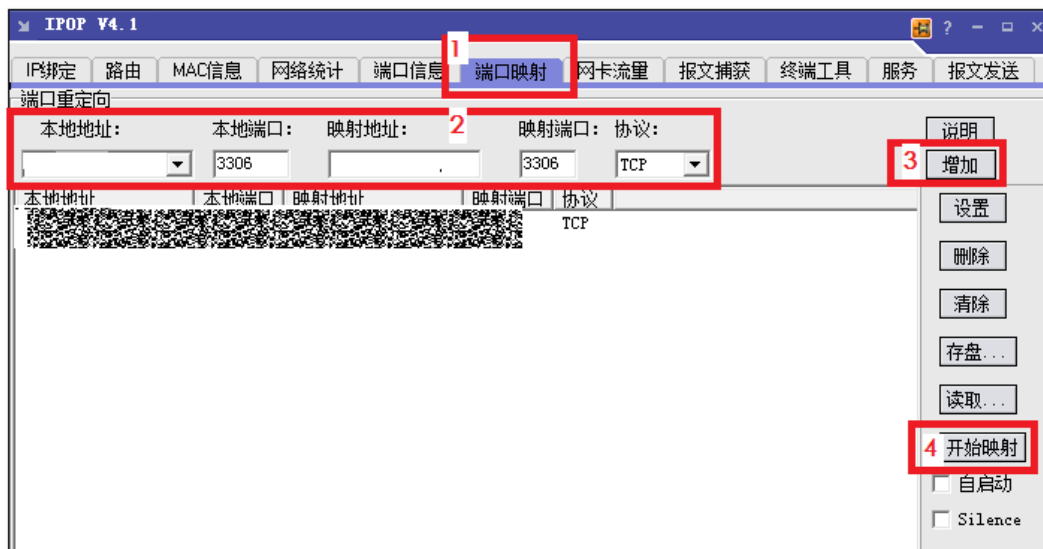
1. 在该机器上ping内网MySQL地址可以ping通，例如：**ping 192.168.1.8**。
2. 在另外一台可上网的机器上ping网关机的公网地址可以ping通，例如ping **202.xx.xx.10**。

**步骤2** 下载端口映射工具IPOP，在网关机上安装IPOP。

**步骤3** 运行端口映射工具，选择“端口映射”，如图2-3所示。

- 本地地址、本地端口：配置为网关机的公网地址和端口（后续在CDM上创建MySQL连接时输入这个地址和端口）。
- 映射地址、映射端口：配置为内网MySQL的地址和端口。

图 2-3 配置端口映射



**步骤4** 单击“增加”，添加端口映射关系。

**步骤5** 单击“开始映射”，这时才会真正开始映射，接收数据包。



至此，就可以在CDM上通过弹性IP读取本地内网MySQL的数据，然后导入到云服务DWS中。

#### 📖 说明

1. CDM要访问本地数据源，也必须给CDM集群配置EIP。
2. 一般云服务DWS默认也是只允许VPC内部访问，创建CDM集群时，必须将CDM的VPC与DWS配置一致，且推荐在同一个内网和安全组，如果不同，还需要配置允许两个安全组之间的数据访问。
3. 端口映射不仅可以用于迁移内网数据库的数据，还可以迁移例如SFTP服务器上的数据。
4. Linux机器也可以通过IPTABLE实现端口映射。
5. 内网中的FTP通过端口映射到公网时，需要检查是否启用了PASV模式。这种情况下客户端和服务端建立连接的时候是走的随机端口，所以除了配置21端口映射外，还需要配置PASV模式的端口范围映射，例如vsftp通过配置pasv\_min\_port和pasv\_max\_port指定端口范围。

----结束

## 2.12 CDM 是否支持参数或者变量？

### 问题描述

CDM是否支持参数或者变量？

### 解决方案

支持。

如果CDM作业使用了在数据开发时配置的**作业参数**或者**变量**，则后续在DataArts Studio数据开发模块调度此节点，可以间接实现CDM作业根据参数变量进行数据迁移。

## 2.13 CDM 迁移作业的抽取并发数应该如何设置？

CDM通过数据迁移作业，将源端数据迁移到目的端数据源中。其中，主要运行逻辑如下：

1. 数据迁移作业提交运行后，CDM会根据作业配置中的“抽取并发数”参数，将每个作业拆分为多个Task，即作业分片。

#### 📖 说明

- 不同源端数据源的作业分片维度有所不同，因此某些作业可能出现未严格按作业“抽取并发数”参数分片的情况。
2. CDM依次将Task提交给运行池运行。根据集群配置管理中的“最大抽取并发数”参数，超出规格的Task排队等待运行。

### 如何调整抽取并发数

1. 集群最大抽取并发数的设置与CDM集群规格有关，并发数上限建议配置为vCPU核数\*2，如**表2-2**所示。

表 2-2 集群最大抽取并发数配置建议

| 规格名称        | vCPUs/内存  | 集群并发数上限参考 |
|-------------|-----------|-----------|
| cdm.large   | 8核 16GB   | 16        |
| cdm.xlarge  | 16核 32GB  | 32        |
| cdm.4xlarge | 64核 128GB | 128       |

图 2-4 集群最大抽取并发数配置



## 2. 作业抽取并发数的配置原则如下：

- 迁移的目的端为文件时，CDM不支持多并发，此时应配置为单进程抽取数据。
- 表中每行数据大小为1MB以下的可以设置多并发抽取，超过1MB的建议单线程抽取数据。
- 作业抽取并发数可参考集群最大抽取并发数配置，但不建议超过集群最大抽取并发数上限。
- 目的端为DLI数据源时，抽取并发数建议配置为1，否则可能会导致写入失败。

图 2-5 作业抽取并发数配置

**任务配置**

作业失败重试 ?

作业分组 ?  添加 编辑 删除

是否定时执行

[隐藏高级属性](#)

**抽取并发数** ?

分片重试次数 ?

是否写入脏数据 ?

开启限速 ?

## 2.14 CDM 是否支持动态数据实时迁移功能？

### 问题描述

CDM是否支持动态数据实时迁移功能？

### 解决方案

不支持。

如果源端在迁移过程中写数据，可能会出现报错。

## 2.15 CDM 是否支持集群关机功能？

### 问题描述

CDM是否支持集群关机功能？

### 解决方案

不支持。

从2022年4月开始，CDM已不再支持集群关机功能。当集群关机时，其底层资源可能会被占用，导致集群可能无法正常开机使用。

## 2.16 如何使用表达式方式获取当前时间？

### 问题描述

如何使用表达式方式获取当前时间？

### 解决方案

您可以在字段映射界面使用DateUtils.format(\${timestamp()},"yyyy-MM-dd HH:mm:ss")表达式获取当前时间，更多表达式设置方式可以参考[表达式转换](#)。

## 2.17 在创建迁移作业时，where 语句参数中的时间格式是怎样的？

### 问题描述

在创建迁移作业时，where语句参数中的时间格式是怎样的？

### 解决方案

请参考Mysql日期、时间字段类型语法特性：<https://dev.mysql.com/doc/refman/8.0/en/datetime.html>。

## 2.18 CDM 作业可以将源表中的字段注释迁移到目标端表吗？

### 问题描述

CDM作业可以将源表中的字段注释迁移到目标端表吗？

### 解决方案

不同版本支持的情况不一样。

具体如下：

2.8.6.1版本支持，2.9.1版本不支持，2.9.2.1版本支持。

## 2.19 如何查询数据连接创建人？

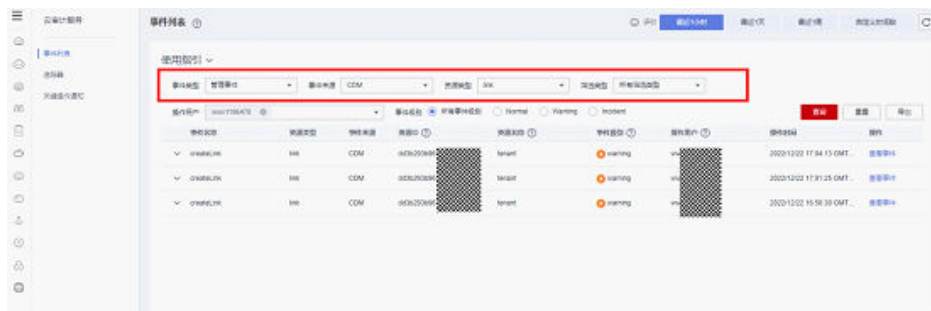
### 问题描述

如何查询数据连接创建人？

## 解决方案

目前有两种方法可以看到。

- 进入云审计服务CTS主界面，事件来源选择“CDM”，资源类型选择“link”，就可以看到所有关于CDM连接管理的操作，更多详情请参见[支持云审计的关键操作](#)。



- 通过华为云开发工具调用API（API Explorer）方式查看连接，详情请参见[查询连接](#)。

```
请求头  请求体
1  {
2  "User-Agent": "API Explorer",
3  "X-Auth-Token": "*****",
4  "Content-Type": "application/json;charset=UTF-8"
5  }
```

```
响应结果
响应头  响应体
1  {
2  "code": "Cdm.0101",
3  "errCode": "Cdm.0101",
4  "message": "Link [hive_connect] does not exist.",
5  "externalMessage": "Link [hive_connect] does not exist."
6  }
```

# 3 故障处理类

## 3.1 日志提示解析日期格式失败时怎么处理？

### 问题描述

在使用CDM迁移其他数据源到云搜索服务（Cloud Search Service）的时候，作业执行失败，日志提示“Unparseable date”，如图3-1所示。

图 3-1 日志提示信息

```
java.text.ParseException: Unparseable date: "2018/01/05 15:15:46"  
    at java.text.DateFormat.parse(DateFormat.java:366) ~[na:1.8.0_112]  
    at org.apache.sqoop.connector.common.DataTypeUtil.convertDateFormat  
    at org.apache.sqoop.connector.elasticsearch.ElasticSearchLoader.toJ  
    at org.apache.sqoop.connector.elasticsearch.ElasticSearchLoader.arr  
7]  
    at org.apache.sqoop.connector.elasticsearch.ElasticSearchLoader.loa
```

### 原因分析

云搜索服务对于时间类型有一个特殊处理：如果存储的时间数据不带时区信息，在Kibana可视化的时候，Kibana会认为该时间为GMT标准时间。

在各个地区会产生日志显示时间与本地时区时间不一致的现象，例如，在东八区某地，日志显示时间比本地时区时间少8个小时。因此在CDM迁移数据到云搜索服务的时候，如果是通过CDM自动创建的索引和类型（例如图3-2中，目的端的“date\_test”和“test1”在云搜索服务中不存在时，CDM会在云搜索服务中自动创建该索引和类型），则CDM默认会将时间类型字段的格式设置为“yyyy-MM-dd HH:mm:ss.SSS Z”的标准格式，例如“2018-01-08 08:08:08.666 +0800”。

图 3-2 作业配置

作业配置

\* 作业名称

---

源连接配置

\* 源连接名称

\* 桶名

\* 源目录或文件

\* 文件格式

[显示高级属性](#)

目的连接配置

\* 目的连接名称

\* 索引

\* 类型

此时，从其他数据源导入数据到云搜索服务时，如果源端数据中的日期格式不完全满足标准格式，例如“2018/01/05 15:15:46”，则CDM作业会执行失败，日志提示无法解析日期格式。需要通过CDM配置字段转换器，将日期字段的格式转换为云搜索服务的目的端格式。

## 解决方案

1. 编辑作业，进入作业的字段的映射步骤，在源端的时间格式字段后面，选择新建转换器，如图3-3所示。

图 3-3 新建转换器

| 源字段 |                         |   | 目的字段                 |                      |                          |                                   |  |
|-----|-------------------------|---|----------------------|----------------------|--------------------------|-----------------------------------|--|
| 列号  | 样值                      | 操作  | 类型                   | 名称                   | 主键                       | 操作                                |  |
| 1   | 913460                  | <input type="button" value="刷新"/> <input type="button" value="删除"/> | keyword              | tripid               | <input type="checkbox"/> | <input type="button" value="删除"/> |  |
| 2   | 765                     | <input type="button" value="刷新"/> <input type="button" value="删除"/> | integer              | duration             | <input type="checkbox"/> | <input type="button" value="删除"/> |  |
| 3   | 2015-08-31 23:26:00.000 | <input type="button" value="刷新"/> <input type="button" value="删除"/> | <input type="text"/> | <input type="text"/> | <input type="checkbox"/> | <input type="button" value="删除"/> |  |

[添加字段+](#)

2. 转换器类型选择“表达式转换”，目前表达式转换支持字符串和日期类型的函数，语法和Java的字符串和时间格式函数非常相似，可以查看[表达式转换](#)了解如何编写表达式。
3. 本例中源时间格式是“yyyy/MM/dd HH:mm:ss”，要将其转换成“yyyy-MM-dd HH:mm:ss.SSS Z”，需要经过如下几步：
  - a. 添加时区信息“+0800”到原始日期字符串的尾部，对应的表达式为：**value + " +0800"**。

- b. 使用原始日期格式来解析字符串，将字符串解析为一个日期对象。可以使用DateUtils.parseDate函数来解析，语法是：**DateUtils.parseDate(String value, String format)**。
- c. 将日期对象格式化成目标格式的字符串，可以使用DateUtils.format函数来格式化，语法是**DateUtils.format(Date date, String format)**。

因此本例中串起来完整的表达式是：

**DateUtils.format(DateUtils.parseDate(value+" +0800","yyyy/MM/dd HH:mm:ss Z"),"yyyy-MM-dd HH:mm:ss.SSS Z")**，如图3-4所示。

图 3-4 配置表达式

## 新建转换器

\* 请选择转换器

\* 表达式

[帮助](#)

4. 保存转换器配置，再保存并运行作业，可解决云搜索服务的解析日期格式失败问题。

## 3.2 字段映射界面无法显示所有列怎么处理？

### 问题描述

在使用CDM从HBase/CloudTable导出数据时，在字段映射界面HBase/CloudTable表的字段偶尔显示不全，无法与目的端字段一一匹配，造成导入到目的端的数据不完整。

### 原因分析

由于HBase/CloudTable无Schema，每条数据的列数不固定，在字段映射界面CDM通过获取样值的方式有较大概率无法获得所有列，此时作业执行完后会造成目的端的数据不全。

这个问题，可以通过以下方法解决：

1. 在CDM的字段映射界面增加字段。
2. 在CDM的作业管理界面直接编辑作业的JSON（修改“fromJobConfig.columns”、“toJobConfig.columnList”这两个参数）。



3. 导出作业的JSON文件到本地，在本地手动修改JSON文件中的参数后（原理同2相同），再导回CDM。

推荐使用方法1，下面以HBase导出到DWS为例进行说明。

### 解决方法一：CDM 的字段映射界面增加字段

1. 获取源端HBase待迁移的表中的所有字段，列族与列之间用“:”分隔，例如：

```
rowkey:rowkey  
g:DAY_COUNT  
g:CATEGORY_ID  
g:CATEGORY_NAME  
g:FIND_TIME  
g:UPLOAD_PEOPLE  
g:ID  
g:INFOMATION_ID  
g:TITLE  
g:COORDINATE_X  
g:COORDINATE_Y  
g:COORDINATE_Z  
g:CONTENT  
g:IMAGES  
g:STATE
```

2. 在CDM的作业管理界面，找到HBase导出数据到DWS的作业，单击作业后面的“编辑”，进入字段映射界面，如图3-5所示。

图 3-5 字段映射 03



3. 单击+添加字段，在弹出框中选择“添加新字段”，如图3-6所示。

图 3-6 添加字段 04



#### 说明

- 添加完字段后，新增的字段在界面不显示样值，这个不影响字段值的传输，CDM会将字段值直接写入目的端。
  - 这里“添加新字段”的功能，要求源端数据源为：MongoDB、HBase、关系型数据库或Redis，其中Redis必须为Hash数据格式。
4. 全部字段添加完之后，检查源端和目的端的字段映射关系是否正确，如果不正确可以拖拽字段调整字段位置。
  5. 单击“下一步”后保存作业。

## 解决方法二：修改 JSON 文件

1. 获取源端HBase待迁移的表中所有的字段，列族与列之间用“:”分隔，例如：

```
rowkey:rowkey
g:DAY_COUNT
g:CATEGORY_ID
g:CATEGORY_NAME
g:FIND_TIME
g:UPLOAD_PEOPLE
g:ID
g:INFOMATION_ID
g:TITLE
g:COORDINATE_X
g:COORDINATE_Y
g:COORDINATE_Z
g:CONTENT
g:IMAGES
g:STATE
```

2. 在DWS目的表中，获取与HBase表对应的字段。

如果DWS目的表中没有HBase对应的字段名，需在DWS表定义中加上，假设DWS表中的字段齐全且如下：

```
rowkey
day_count
category
category_name
find_time
upload_people
```

```
id
information_id
title
coordinate_x
coordinate_y
coordinate_z
content
images
state
```

- 在CDM的作业管理界面，找到HBase到DWS的作业，选择作业后面的“更多 > 编辑作业JSON”。
- 在CDM界面编辑作业的JSON文件。

- 修改源端的“fromJobConfig.columns”参数，配置为1获取的HBase的字段，列号之间使用“&”分隔，列族与列之间用“:”分隔，如下：

```
"from-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "fromJobConfig.table",
          "value": "HBase"
        },
        {
          "name": "fromJobConfig.columns",
          "value":
"rowkey:rowkey&g:DAY_COUNT&g:CATEGORY_ID&g:CATEGORY_NAME&g:FIND_TIME&g:UP
LOAD_PEOPLE&g:ID&g:INFOMATION_ID&g:TITLE&g:COORDINATE_X&g:COORDINATE_Y&g:
COORDINATE_Z&g:CONTENT&g:IMAGES&g:STATE"
        }
      ],
      "name": "fromJobConfig"
    }
  ]
}
```

- 修改目的端的“toJobConfig.columnList”参数，配置为2中DWS的字段列表。

这里的顺序必须与HBase保持一致，才能保证正确的字段映射关系，字段名之间使用“&”分隔，如下：

```
"to-config-values": {
  "configs": [
    {
      "inputs": [
        {
          "name": "toJobConfig.schemaName",
          "value": "dbadmin"
        },
        {
          "name": "toJobConfig.tablePreparation",
          "value": "DO_NOTHING"
        },
        {
          "name": "toJobConfig.tableName",
          "value": "DWS "
        },
        {
          "name": "toJobConfig.columnList",
          "value":
"rowkey&day_count&category&category_name&find_time&upload_people&id&infomation_"
        }
      ]
    }
  ]
}
```

```
id&title&coordinate_x&coordinate_y&coordinate_z&content&images&state"
    },
    {
      "name": "toJobConfig.shouldClearTable",
      "value": "true"
    }
  ],
  "name": "toJobConfig"
}
]
```

- c. 其他参数保持不变，单击“保存并运行”。
5. 作业完成后，查询DWS表中的数据是否和HBase中的数据匹配。如果不匹配，请检查JSON文件中HBase和DWS字段的顺序是否一致。

## 3.3 CDM 迁移数据到 DWS 时如何选取分布列？

### 问题描述

CDM迁移数据到DWS时如何选取分布列？

### 解决方案

在使用CDM迁移数据到数据仓库服务（DWS）或者FusionInsight LibrA，且CDM在DWS端自动创建一个新表时，在创建作业的字段映射界面，需要选择分布列，如图3-7所示。

图 3-7 选取分布列

| 源字段     |    |             |      | 目的字段    |             |                          |    |
|---------|----|-------------|------|---------|-------------|--------------------------|----|
| 名称      | 样值 | 类型          | 操作   | 名称      | 类型          | 分布列                      | 操作 |
| COLUMN1 | 1  | VARCHAR(50) | 🔄 🗑️ | COLUMN1 | VARCHAR(50) | <input type="checkbox"/> | 🗑️ |
| COLUMN2 | LU | VARCHAR(50) | 🔄 🗑️ | COLUMN2 | VARCHAR(50) | <input type="checkbox"/> | 🗑️ |
| COLUMN3 | 15 | VARCHAR(50) | 🔄 🗑️ | COLUMN3 | VARCHAR(50) | <input type="checkbox"/> | 🗑️ |

由于分布列的选取，对于DWS/FusionInsight LibrA的运行非常重要，在CDM数据迁移到DWS/FusionInsight LibrA过程中，建议按如下顺序选取分布列：

1. 有主键可以使用主键作为分布列。
2. 多个数据段联合做主键的场景，建议设置所有主键作为分布列。
3. 在没有主键的场景下，如果没有选择分布列，DWS会默认第一列作为分布列，可能会有数据倾斜风险。

因此，在单表或整库导入到DWS/FusionInsight LibrA时，建议您在此处手动选择分布列，如果您没有选择，CDM会自动选择一个分布列。关于分布列的更多信息，请参见[数据仓库服务](#)。

DWS主键或表只有一个字段时，要求字段类型必须是如下常用的字符串、数值、日期类型。从其他数据库迁移到DWS时，如果选择自动建表，主键必须为以下类型，未设

置主键的情况下至少要有一个字段是以下类型，否则会无法创建表导致CDM作业失败。

- INTEGER TYPES: TINYINT, SMALLINT, INT, BIGINT, NUMERIC/DECIMAL
- CHARACTER TYPES: CHAR, BPCHAR, VARCHAR, VARCHAR2, NVARCHAR2, TEXT
- DATA/TIME TYPES: DATE, TIME, TIMETZ, TIMESTAMP, TIMESTAMPTZ, INTERVAL, SMALLDATETIME

## 3.4 迁移到 DWS 时出现 value too long for type character varying 怎么处理?

### 问题描述

在使用CDM迁移数据到数据仓库服务（DWS）或者FusionInsight LibrA时，如果迁移作业失败，且执行日志中出现“value too long for type character varying”错误提示，如[图3-8](#)所示。

图 3-8 日志信息

```
Caused by: org.postgresql.util.PSQLException: ERROR: value too long for type character varying(50)
  Where: COPY fl_behavior_module, line 72, column MODULE_NAME: "点正以 融科 融数 - 融平 融理 融以 融信"
    at org.postgresql.core.v3.QueryExecutorImpl.receiveErrorResponse(QueryExecutorImpl.java:2477)
    at org.postgresql.core.v3.QueryExecutorImpl.processCopyResults(QueryExecutorImpl.java:1107)
    at org.postgresql.core.v3.QueryExecutorImpl.writeToCopy(QueryExecutorImpl.java:989)
    at org.postgresql.core.v3.CopyInImpl.writeToCopy(CopyInImpl.java:35)
    ... 16 common frames omitted
```

### 原因分析

这种情况一般是在迁移到DWS时数据有中文，且创建作业时选择了目的端自动建表的情况下。原因是DWS的varchar类型是按字节计算长度，一个中文字符在UTF-8编码下可能要占3个字节。当中文字符的字节超过DWS的varchar的长度时，就会出现错误：value too long for type character varying。

### 解决方案

这个问题，可以通过将目的端作业参数“扩大字符字段长度”选择“是”来解决，选择此选项后，再创建目的表时会自动将varchar类型的字段长度扩大3倍。

编辑CDM的表/文件迁移作业，目的端作业配置下“自动创表”选择“不存在时创建”，则高级属性下面会出现参数“扩大字符字段长度”，配置该参数为“是”即可，如[图3-9](#)所示。

图 3-9 扩大字符字段长度

**目的端作业配置**

|                       |  |  |
|-----------------------|--|--|
| * 目的连接名称              | <input type="text" value="dwslink"/>                       | <input type="button" value="新建连接"/>  |
| * 模式或表空间 <sup>?</sup> | <input type="text"/>                                       | <input style="border: none; background: none;" type="button" value="..."/> |
| 自动创表 <sup>?</sup>     | <input type="text" value="不存在时创建"/>                        |  |
| * 表名 <sup>?</sup>     | <input type="text"/>                                       | <input style="border: none; background: none;" type="button" value="..."/> |
| 是否压缩 <sup>?</sup>     | <input type="radio"/> 是 <input checked="" type="radio"/> 否 |  |
| 存储模式 <sup>?</sup>     | <input type="text" value="行模式"/>                           |  |
| 导入前清空数据 <sup>?</sup>  | <input type="radio"/> 是 <input checked="" type="radio"/> 否 |  |
| 隐藏高级属性                |  |  |
| 先导入阶段表 <sup>?</sup>   | <input type="radio"/> 是 <input checked="" type="radio"/> 否 |  |
| 扩大字符字段长度 <sup>?</sup> | <input checked="" type="radio"/> 是 <input type="radio"/> 否 |  |

## 3.5 OBS 导入数据到 SQL Server 时出现 Unable to execute the SQL statement 怎么处理?

### 问题描述

使用CDM从OBS导入数据到SQL Server时，作业运行失败，错误提示为：Unable to execute the SQL statement. Cause : 将截断字符串或二进制数据。

### 原因分析

用户OBS中的数据超出了SQL Server数据库的字段长度限制。

### 解决方案

在SQL Server数据库中建表时，将数据库字段改大，长度不能小于源端OBS中的数据长度。

## 3.6 获取集群列表为空/没有权限访问/操作时报当前策略不允许执行?

### 问题描述

在使用CDM时，可能遇到如下权限相关的问题：

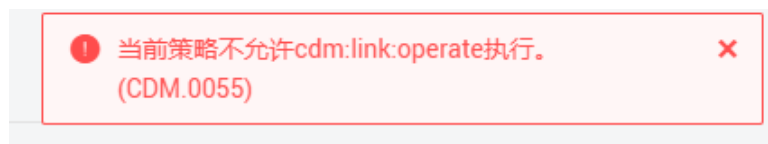
- 跳转到CDM首页，获取到的集群列表为空。
- 提示没有权限访问，如[图3-10](#)所示。
- 执行启动作业/重启集群等操作时，报错当前策略不允许执行，如[图3-11](#)所示。

图 3-10 没有权限访问



很抱歉，您没有访问权限。  
请联系您的账号管理员开通权限。

图 3-11 不允许创建连接



### 原因分析

以上所列的问题均属于权限配置问题。

### 解决方案

- 如果是作为DataArts Studio服务CDM组件使用：
  - a. 检查用户是否添加DAYU Administrator或DAYU User角色，参考[DataArts Studio权限管理](#)。
  - b. 是否有对应工作空间的权限，如开发者、访客等，参考[DataArts Studio权限列表](#)。
- 如果是独立CDM服务使用：
  - a. 检查是否开启IAM细粒度鉴权。
    - 如果未开启，检查用户组是否添加CDM Administrator角色。

- 如果已开启，请继续执行**步骤2**继续检查。
- b. 检查用户是否添加CDM访问策略，包含自定义策略或预设策略，如CDM FullAccess、CDM ReadOnlyAccess等，参考**CDM权限管理**。
- c. 检查对应企业项目是否添加拒绝访问策略。

## 3.7 Oracle 迁移到 DWS 报错 ORA-01555

### 问题描述

使用CDM迁移Oracle数据至DWS，报错**图3-12**所示。

图 3-12 报错现象

```
665 2020-09-21 22:51:02,991 ERROR LocalJobRunner Map Task #3 [org.apache.sqoop.common.SqoopException:111] SqoopException
666 java.sql.SQLException: ORA-01555: snapshot too old: rollback segment number 3 with name "_SYS$SMU3_20976775319" too small
667
668   at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:494)
669   at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:446)
670   at oracle.jdbc.driver.T4C8Oall.processERROR(T4C8Oall.java:1054)
671   at oracle.jdbc.driver.T4CTTIfun.receive(T4CTTIfun.java:623)
672   at oracle.jdbc.driver.T4CTTIfun.doRPC(T4CTTIfun.java:252)
673   at oracle.jdbc.driver.T4C8Oall.doOALL(T4C8Oall.java:612)
674   at oracle.jdbc.driver.T4CPreparedStatement.doOall8(T4CPreparedStatement.java:226)
675   at oracle.jdbc.driver.T4CPreparedStatement.fetch(T4CPreparedStatement.java:1023)
676   at oracle.jdbc.driver.OracleStatement.fetchMoreRows(OracleStatement.java:3353)
677   at oracle.jdbc.driver.InsensitiveScrollableResultSet.fetchMoreRows(InsensitiveScrollableResultSet.java:736)
678   at oracle.jdbc.driver.InsensitiveScrollableResultSet.absoluteInternal(InsensitiveScrollableResultSet.java:692)
679   at oracle.jdbc.driver.InsensitiveScrollableResultSet.next(InsensitiveScrollableResultSet.java:406)
680   at org.apache.sqoop.connector.jdbc.sql.impl.WrapResultSet.next(WrapResultSet.java:36)
681   at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extractObjectRecord(GenericJdbcExtractor.java:151)
682   at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:129)
683   at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:59)
684   at org.apache.sqoop.job.mapred.SqoopMapper.runInternal(SqoopMapper.java:184)
685   at org.apache.sqoop.job.mapred.SqoopMapper.run(SqoopMapper.java:81)
686   at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:799)
687   at org.apache.hadoop.mapred.MapTask.run(MapTask.java)
688   at org.apache.hadoop.mapred.LocalJobRunner$Job$MapTaskRunnable.run(LocalJobRunner.java:271)
689   at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
690   at java.util.concurrent.FutureTask.run(FutureTask.java:266)
691   at org.apache.sqoop.submission.mapreduce.MapperExecutorGroup$1.lambda$execute$0(MapperExecutorGroup.java:222)
692   at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
693   at java.util.concurrent.FutureTask.run(FutureTask.java:266)
694   at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
695   at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
696   at java.lang.Thread.run(Thread.java:748)
697 Caused by: oracle.jdbc.OracleDatabaseException: ORA-01555: snapshot too old: rollback segment number 3 with name "_SYS$SMU3_20976775319" too small
698
699   at oracle.jdbc.driver.T4CTTIoer11.processERROR(T4CTTIoer11.java:498)
700   ... 28 common frames omitted
```

### 原因分析

1. 数据迁移，整表查询且该表数据量大，那么查询时间较长。
2. 查询过程中，其他用户频繁进行commit操作。
3. Oracle的RBS（rollbackpace 回滚时使用的表空间）较小，造成迁移任务没有完成，源库已更新，回滚超时。

### 建议与总结

1. 调小每次查询的数据量。
2. 通过修改数据库配置调大Oracle的RBS。



## 3.8 MongoDB 连接迁移失败时如何处理？

### 问题描述

MongoDB连接迁移失败时如何处理？

### 解决方案

在默认情况下，userAdmin角色只具备对角色和用户的管理，不具备对库的读和写权限。

当用户选择MongoDB连接迁移失败时，用户需查看MongoDB连接中用户的权限信息，确保对指定库具备ReadWrite权限。

## 3.9 Hive 迁移作业长时间卡顿怎么办？

### 问题描述

Hive迁移作业长时间卡顿怎么办？

### 解决方案

为避免Hive迁移作业长时间卡顿，可手动停止迁移作业后，通过编辑Hive连接增加如下属性设置：

- 属性名称：hive.server2.idle.operation.timeout
- 值：10m

如图所示：



## 3.10 使用 CDM 迁移数据由于字段类型映射不匹配导致报错怎么处理？

### 问题描述

在使用CDM迁移数据到数据仓库服务（DWS）时，迁移作业失败，且执行日志中出现“value too long for type character varying”错误提示。

### 原因分析

这种情况一般是源表与目标表类型不匹配导致，例如源端dli字段为string类型，目标端dws字段为varchar(50)类型，导致精度缺省，就会报：value too long for type character varying。类似的问题还有string转bigint，bigint转int。

## 解决方案

- 根据报错信息找到哪个字段映射有问题，找DBA修改表结构。
- 如果只有极少数据有问题，可以配置脏数据策略解决。

## 3.11 MySQL 迁移时报错“JDBC 连接超时”怎么办？

### 问题描述

MySQL迁移时报错：Unable to connect to the database server. Cause: connect timed out.

### 原因分析

这种情况是由于表数据量较大，并且源端通过where语句过滤，但并非索引列，或列值不离散，查询会全表扫描，导致JDBC连接超时。

图 3-13 非索引列

The screenshot shows two configuration panels for a migration job. The left panel, titled '源端作业配置' (Source Job Configuration), includes fields for '源连接名称' (Source Connection Name) set to 'mysql', '使用SQL语句' (Use SQL Statement) set to '是' (Yes), '模式或表空间' (Schema or Tablespace) set to 'SQOOP', '表名' (Table Name) set to 'rf\_BaoWeiFu\_test\_sql\_To', and a 'Where子句' (Where Clause) containing 'C\_date > '2021-02-27 10:43:04.123''. The right panel, titled '目的端作业配置' (Target Job Configuration), includes fields for '目的连接名称' (Target Connection Name) set to 'dli', '资源队列' (Resource Queue) set to 'dli\_notdelete', '数据库名称' (Database Name) set to 'abcd', and '表名' (Table Name) set to 'dddd'. Both panels have a '导入脏空数据' (Import Dirty/Empty Data) checkbox set to '是' (Yes).

## 解决方案

1. 优先联系DBA修改表结构，将需要过滤的列配置为索引列，然后重试。  
如果由于数据不离散，导致还是失败请参考2~4，通过增大JDBC超时时间解决。
2. 根据作业找到对应的MySQL连接名称，查找连接信息。

图 3-14 连接信息

The screenshot shows a table with two columns: '名称' (Name) and '连接信息' (Connection Information). The first row has a checkbox, the name 'mysql2dli', and the connection information 'mysql-dli'. The 'mysql-dli' text is highlighted with a red box.

| <input type="checkbox"/> | 名称        | 连接信息      |
|--------------------------|-----------|-----------|
| <input type="checkbox"/> | mysql2dli | mysql-dli |

3. 单击“连接管理”，在“操作”列中，单击“连接”进行编辑。

图 3-15 连接



4. 打开高级属性，在“连接属性”中建议新增“connectTimeout”与“socketTimeout”参数及参数值，单击“保存”。

图 3-16 编辑高级属性



## 3.12 创建了 Hive 到 DWS 类型的连接，进行 CDM 传输任务失败时如何处理？

### 问题描述

创建了Hive到DWS类型的连接，进行CDM传输任务失败。

### 解决方案

建议清空历史数据后再次尝试该任务。在使用CDM迁移作业的时候需要配置清空历史数据，然后再做迁移，可大大降低任务失败的概率。

## 3.13 如何使用 CDM 服务将 MySQL 的数据导出成 SQL 文件，然后上传到 OBS 桶？

### 问题描述

如何使用CDM服务将MySQL的数据导出成SQL文件，然后上传到OBS桶？

## 解决方案

CDM服务暂不支持该操作，建议通过手动导出MySQL的数据文件，然后在服务器上开启SFTP服务，然后新建CDM作业，源端是SFTP协议，目的端是OBS，将文件传过去。

## 3.14 如何处理 CDM 从 OBS 迁移数据到 DLI 出现迁移中断失败的问题？

### 问题描述

CDM从OBS迁移数据到DLI出现迁移中断失败。

### 解决方案

此类作业问题表现为配置了脏数据写入，但并无脏数据。这种情况下需要调低并发任务数，即可避免此类问题。

## 3.15 报错“配置项[linkConfig.createBackendLinks]不存在”或“配置项 [throttlingConfig.concurrentSubJobs] 不存在怎么办”？

### 问题描述

创建数据连接时报错“配置项[linkConfig.createBackendLinks]不存在”或创建作业时报错“配置项 [throttlingConfig.concurrentSubJobs] 不存在”。

### 原因分析

当同时存在多个不同版本的集群，先在低版本CDM集群创建数据连接或保存作业后，再进入高版本CDM集群时，会偶现此类故障。

### 解决方案

需手动清理浏览器缓存，即可避免此类问题。

## 3.16 新建 MRS Hive 连接时，提示：CORE\_0031:Connect time out. (Cdm.0523) 怎么解决？

### 问题描述

新建MRS Hive连接时，提示：CORE\_0031:Connect time out. (Cdm.0523)。

### 解决方案

新建MRS Hive连接时，提示无法下载配置文件，实际是用户权限不足。建议您新建一个业务用户，给对应的权限后重试即可。

如果要创建MRS安全集群的数据连接，不能使用admin用户。因为admin用户是默认的管理页面用户，这个用户无法作为安全集群的认证用户来使用。您可以创建一个新的MRS用户，然后在创建MRS数据连接时，“用户名”和“密码”填写为新建的MRS用户及其密码。

#### 📖 说明

- 如果CDM集群为2.9.0版本及之后版本，且MRS集群为3.1.0及之后版本，则所创建的用户至少需具备Manager\_viewer的角色权限才能在CDM创建连接；如果需要对MRS组件的库、表、列进行操作，还需要参考MRS文档添加对应组件的库、表、列操作权限。
- 如果CDM集群为2.9.0之前的版本，或MRS集群为3.1.0之前的版本，则所创建的用户需要具备Manager\_administrator或System\_administrator权限，才能在CDM创建连接。
- 仅具备Manager\_tenant或Manager\_auditor权限，无法创建连接。

## 3.17 迁移时已选择表不存在时自动创表，提示“CDM not support auto create empty table with no column”怎么处理？

### 问题描述

迁移时已选择表不存在时自动创表，提示“CDM not support auto create empty table with no column”。

### 原因分析

这是由于数据库表名中含有特殊字符导致识别出语法错误，按数据库对象命名规则重新命名后恢复正常。

例如，DWS数据仓库中的数据表命名需要满足以下约束：长度不超过63个字符，以字母或下划线开头，中间字符可以是字母、数字、下划线、\$、#。

### 解决方案

重新命名。

## 3.18 创建 Oracle 关系型数据库迁移作业时，无法获取模式名怎么处理？

### 问题描述

创建Oracle关系型数据库迁移作业时，无法获取模式名。

### 原因分析

这是由于可能上传了暂不支持的最新ORACLE\_8驱动（如Oracle Database 21c (21.3) drivers）。

## 解决方案

推荐使用Oracle Database 12c中的ojdbc8.jar驱动，下载地址：<https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html>。

## 3.19 MySQL 迁移时报错：invalid input syntax for integer: "true"

### 问题描述

数据库中存储的是1或0，但没有true和false的数据，但MySQL迁移时读取到的是true或false，提示报错信息：Unable to execute the SQL statement. Cause: ERROR: invalid input syntax for integer: "true" Where: COPY sd\_mask\_ext, line 1, column mask\_type.

### 原因分析

MySQL默认开启配置tinyInt1isBit=true，会将TINYINT(1)当作BIT也就是Types.BOOLEAN来处理，将1或0读取为true或false。

### 解决方案

在MySQL数据连接高级属性中，连接属性新增如下参数之一即可，这样就可以在目的端正常建表。

- “tinyInt1isBit” 参数，参数值设为“false”。
- “mysql.bool.type.transform” 参数，参数值设为“false”。

图 3-17 添加连接属性

隐藏高级属性

一次请求行数 ①

一次提交行数 ①

SSL加密 ②  是  否

| 属性名称                 | 值                    | 操作 |
|----------------------|----------------------|----|
| <input type="text"/> | <input type="text"/> | 删除 |

连接属性 ②

+ 添加

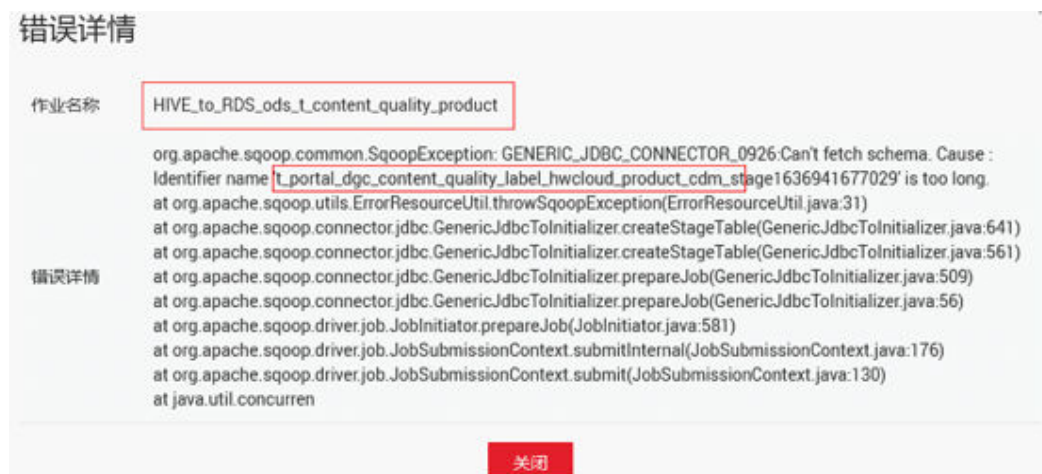
## 3.20 作业源端是 Oracle 时，运行时间过长报 snapshot too old 怎么解决？

是Oracle的约束限制导致，迁移过程中源端表中所有数据不能存在更新、删除和新增操作。可以加大UNDO\_RETENTION，同时调整UNDO表空间大小即可。

## 3.21 整库迁移到 Hive，报错 Identifier name is too long 如何处理？

### 问题描述

迁移任务报错表名太长，但表名实际没有这么长。



### 原因分析

在任务迁移时，导入数据前会先创建一个实际表名+阶段表后缀的阶段表，最终导致的作业异常。

### 解决方案

- 在作业配置高级属性将导入阶段表设置为否，这样就不会先导入阶段表。
- 缩短实际表的表名长度。

## 3.22 迁移数据到 DLI 时有数据丢失怎么处理？

### 问题描述

目的端是DLI的迁移作业执行成功后，显示迁移的数据条数和DLI表中查询的数量对不上，存在数据丢失。

### 原因分析

1. 查看客户的作业配置，客户并发数设置的是3，如图所示。

### 任务配置

作业失败重试 ②

作业分组 ②  [添加](#) [编辑](#) [删除](#)

是否定时执行

[隐藏高级属性](#)

抽取并发数 ②

是否写入脏数据 ②

- 查看DLI官网文档不建议同时对一张表并发插入数据。

### 注意事项

- 表必须已经存在。
- 如果动态分区不需要指定分区，则将“part\_spec”作为普通字段放置SELECT语句中。
- 被插入的OBS表在建表时只能指定文件夹路径。
- 源表和目标表的数据类型和列字段个数应该相同，否则插入失败。
- 不建议对同一张表并发插入数据，因为有一定概率发生并发冲突，导致插入失败。
- INSERT INTO命令用于将查询的结果追加到目标表中。

## 解决方案

将作业的抽取并发数改成1，重跑作业问题解决。

## 3.23 创建 Oracle 数据连接测试连通性成功，连接管理界面中测试连接失败。是什么原因？

### 问题描述

创建Oracle数据连接，创建连接时测试连通性成功。

在连接管理界面中，测试Oracle数据连接失败。提示如下信息：

“无法连接服务器，请检查IP、主机名、端口填写是否正确，检查网络安全组和防火墙配置是否正确，参考数据库返回消息 ORA-01005 null password given. logon denied(Cdm 0941)”

### 解决方案

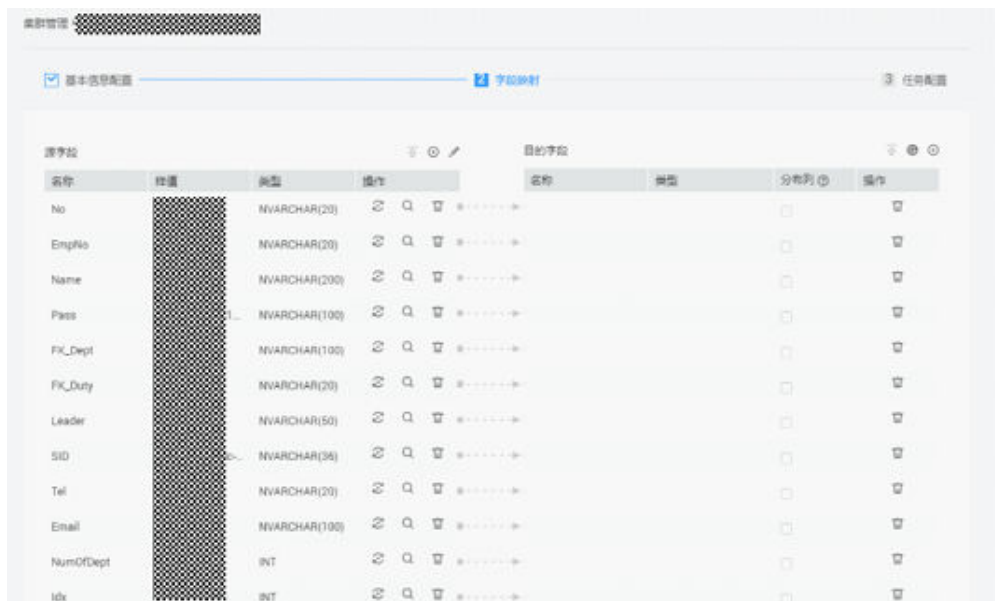
请检查IP、主机名、端口填写是否正确，检查网络安全组和防火墙配置是否正确，参考数据库返回消息进行定位，发现设置Oracle数据库密码少于8个字符，然后再创建数据连接问题解决。



## 3.24 作业配置表不存在时自动创建，目的端字段映射不出来怎么处理？

### 问题描述

迁移SQL Server数据到DWS，目的端配置了当表不存在时自动创建，目的端字段映射不出来，如下图所示。



### 原因分析

1. 查看后端日志报：org.postgresql.util.PSQLException: ERROR: relation "表名" does not exist.

```
2021-10-29 14:40:31,064 ERROR pool-3148-thread-1 [org.apache.sqoop.connector.jdbc.configuration.LinkConfiguration:336] An error occurred when obtaining the fields by table.
org.postgresql.util.PSQLException: ERROR: relation "dwi_comm.port_emp" does not exist
Position: 15
at org.postgresql.core.v3.QueryExecutorImpl.receiveErrorResponse(QueryExecutorImpl.java:2552)
at org.postgresql.core.v3.QueryExecutorImpl.processResults(QueryExecutorImpl.java:2284)
at org.postgresql.core.v3.QueryExecutorImpl.execute(QueryExecutorImpl.java:322)
at org.postgresql.jdbc.PgStatement.executeInternal(PgStatement.java:481)
at org.postgresql.jdbc.PgStatement.execute(PgStatement.java:491)
at org.postgresql.jdbc.PgPreparedStatement.executeWithFlags(PgPreparedStatement.java:164)
at org.postgresql.jdbc.PgPreparedStatement.executeQuery(PgPreparedStatement.java:114)
at org.apache.sqoop.connector.jdbc.sql.impl.WrapPreparedStatement.lambda$executeQuery$0(WrapPreparedStatement.java:28)
at java.security.AccessController.doPrivileged(Native Method)
at org.apache.sqoop.util.JdbcSandbox.doPrivileged(JdbcSandbox.java:41)
at org.apache.sqoop.connector.jdbc.sql.impl.WrapPreparedStatement.executeQuery(WrapPreparedStatement.java:28)
at sun.reflect.GeneratedMethodAccessor60.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.apache.sqoop.handler.LinkMetaRequestHandler.getField(LinkMetaRequestHandler.java:572)
at org.apache.sqoop.handler.LinkMetaRequestHandler.getLinkMetaData(LinkMetaRequestHandler.java:240)
at org.apache.sqoop.handler.LinkMetaRequestHandler.access$500(LinkMetaRequestHandler.java:67)
at org.apache.sqoop.handler.LinkMetaRequestHandler$1.call(LinkMetaRequestHandler.java:85)
at org.apache.sqoop.handler.LinkMetaRequestHandler$1.call(LinkMetaRequestHandler.java:85)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)
```

2. 怀疑是CDM 集群开启沙箱所导致的，后端对集群取消沙箱，重启CDM 集群后问题依然存在。
3. 查看CDM迁移作业，源端数据库表名全部为小写，但是迁移的表中是包含有大写字母，将所要迁移的表名跟数据库中保持一致，目的端字段就可以映射出来了，问题解决。

## 解决方案

在作业设置中，源端配置中迁移的数据库表名应按照数据库中的名称填写或者通过搜索选择表名，问题解决。

### 3.25 作业从旧集群导出，再导入到新的集群失败怎么解决？

#### 问题描述

旧CDM集群是2.6.0版本，新集群是2.8.6.1版本，导入作业报错如下图所示。



#### 原因分析

1. 初步怀疑是新老集群部分参数修改不兼容导致的，通过查看老集群导出的作业json文件，包含throttlingConfig.concurrentSubJobs参数（并发子作业数，新集群已取消这个配置项）。

```
},
"configs":[{
  "inputs":[{
    "name":"throttlingConfig.concurrentSubJobs",
    "value":"10"
  }],
  {
    "name":"throttlingConfig.numExtractors",
    "value":"1"
  },
  {
    "name":"throttlingConfig.submitToCluster",
    "value":"false"
  }
],
}
```

2. 让客户在导出的json文件中删除以下配置项，重新导入作业到新集群，导入成功。

```
{  
  "name": "throttlingConfig.concurrentSubJobs",  
  "value": "10"  
},
```

## 解决方案

将导出的作业json文件中 "name": "throttlingConfig.concurrentSubJobs" 配置项删除后重新导入作业json即可。

## 3.26 迁移 HDFS 文件，报错无法获取块怎么处理？

### 问题描述

1. 用户HDFS为线下自建的，往OBS迁移文件建立好连接器后，测试源端和目的端的连通性都是没问题的。
2. 任务启动时报如下错误：

```
Error: java.io.IOException: org.apache.hadoop.hdfs.BlockMissingException: Could not obtain  
block: BP-787476470-192.168.152.10-1573351961380:blk_1073742171_1347 file=/user/hive/  
warehouse/db_hive.db/emp/emp.txt (state=,code=0)
```

### 原因分析

1. 使用HDFS客户端get文件可以正常获取，所以不是文件块丢失。
2. 查看HDFS服务的所有DataNode实例是否都已启动，此时DataNode状态为停止会获取不到块，以及cdm和DataNode节点的网络是否正常。  
注：9866端口是HDFS文件系统DataNode的数据传输接口。

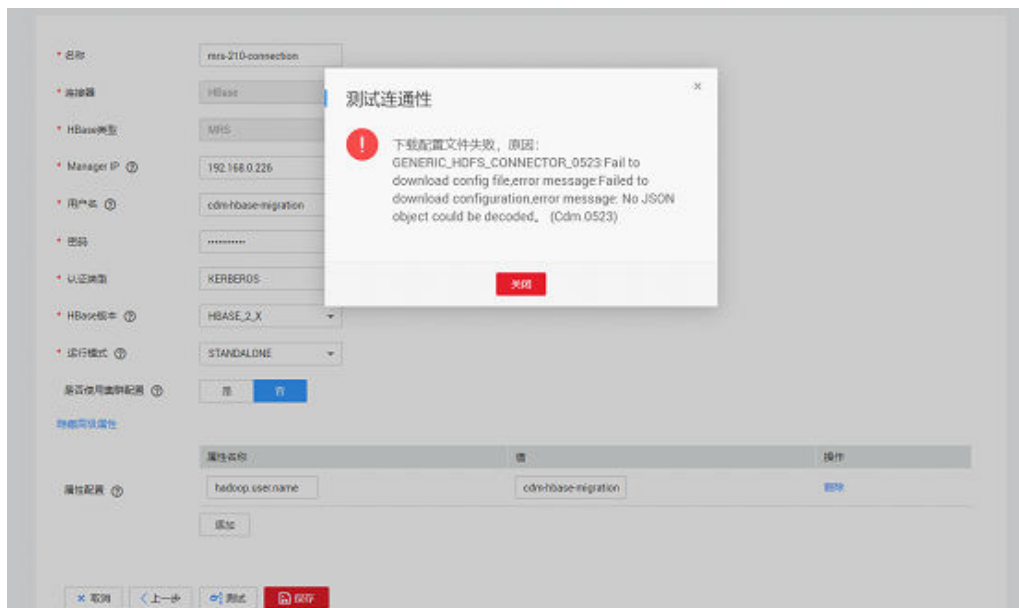
## 解决方案

因为DataNode节点防火墙为开启状态，CDM在与Datanode建立连接时失败导致获取块失败。关闭Datanode节点的防火墙后问题解决。

## 3.27 创建 MRS 数据连接时测试连通性报下载配置文件失败错误怎么处理？

### 问题描述

创建MRS连接器报如下错误，在Manager检查用户权限是有的，检查网络也都是通的。



## 解决方案

登录Manager初始化用户后，重新建立数据连接即可。

## 3.28 CDM 作业管理访问不了，提示网络或服务器访问异常怎么处理？

### 问题描述

作业管理页面访问不了，提示“网络或服务器异常，请重试”的报错。

### 解决方案

1. F12看下接口返回都正常。
2. 查看CDM集群各项指标是否正常：如磁盘、内存、CPU。
3. 如果CDM集群以上指标都正常，用户侧清理浏览器缓存之后重新单击作业管理即可。

## 3.29 通过 CDM 从 OBS 迁移数据到 DLI，同样的作业在新版本集群迁移失败？

### 问题描述

客户通过CDM从OBS迁移到DLI，使用两个集群分别迁移，源端和目标端以及作业配置都一样，2.6.0版本的CDM集群作业可以迁移成功，2.8.6版本的集群迁移失败。报错作业日志如下图所示。

```
2021-06-29 17:55:15.240.FBROD:query-loader-0 [org.apache.sqoop.connector.uquery.processor.Dataconsumer:174] failed to convert the record.
java.lang.NumberFormatException: For input string: ""
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
    at java.lang.Integer.parseInt(Integer.java:592)
    at java.lang.Integer.parseInt(Integer.java:615)
    at org.apache.sqoop.connector.uquery.processor.Dataconsumer.convert(Dataconsumer.java:297)
    at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:172)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
```

## 原因分析

1. 初步怀疑是源端和目标端在字段类型转换出现异常。

|    |                              |                           |         |
|----|------------------------------|---------------------------|---------|
| 15 |                              | NETWORK_NODE_TYPE         | string  |
| 16 |                              | REGION_TYPE               | string  |
| 17 |                              | SOLUTION_TYPE             | string  |
| 18 |                              | GLOBAL_SERVICE_SCALE      | string  |
| 19 |                              | EXTERNAL_GLOBAL_DOMAIN_N  | string  |
| 20 |                              | MGMT_NETWORK_TYPE         | string  |
| 21 |                              | GLOBAL_DOMAIN_NAME_POSTF  | string  |
| 22 |                              | SOLUTION_SCENSE           | string  |
| 23 | TRUE                         | IS_LOCAL                  | boolean |
| 24 | unknown                      | MGMT_NODE_CPU_ARCH        | string  |
| 25 |                              | THREE_DC_IN_TWO_PLACES    | boolean |
| 26 |                              | PREVIOUS_SOLUTION_VERSION | string  |
| 27 |                              | HISTORY_VERSION_LIST      | string  |
| 28 |                              | SUPPORT_AZ_HA             | boolean |
| 29 | 1611884582334                | LAST_MODIFIED             | bigint  |
| 30 | 9b7b0904-26ce-4129-811c-945d | CLOUD_ID                  | string  |
| 31 | 1622270095                   | UPDATED_AT                | bigint  |
| 32 | 1622270095                   | CREATED_AT                | bigint  |

2. 将目标端表字段类型bigint改为 string，重新跑作业还是失败，报错内容跟之前一样。
3. 配置开启脏数据，重跑作业后作业依旧失败，但是有3条数据已迁移到目标表。
4. 通过对比迁移失败的数据记录和成功的距离，怀疑是类型为boolean的字段导致的。

## 解决方案

将目标端boolean类型字段修改成string 后作业跑成功，因为客户源端boolean类型的字段有空值，从而导致迁移失败。

2.8.6版本CDM集群校验更严格，在处理boolean类型数据的逻辑是：目标端的数据类型是boolean类型，会在插入的时候会去检查，不是true/false就会报错。

## 3.30 CDM 迁移 DWS 数据报错 Read timedout 怎么处理？

### 问题描述

客户使用cdm迁移DWS A集群表数据到DWS B集群，成功写入部分后报错：

```
org.postgresql.util.PSQLException: Database connection failed when ending copy.....caused by:  
java.net.SocketTimeoutException:Read timed out;
```

## 故障分析

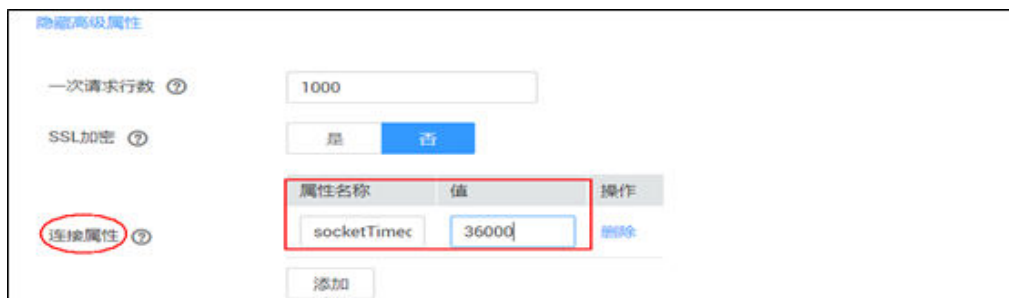
作业配置中源端目标端均通过where语句多条件过滤，并非索引列，查询会全表扫描，且数据量在上亿行，数据量庞大，导致JDBC数据库连接失败，读取数据超时，进而导致sqoop异常，作业失败。

迁移作业是CDM作为客户端先从源数据中抽取部分数据，写到目标端，再进行下一次部分数据抽取，写入目标端，往复执行，直到抽取到写入完成。因此可以添加高级属性：socketTimeout 参数，保证在每次抽取写入数据间隔，CDM一直保持正常会话。

## 解决方案

通过增大jdbc连接超时时间的控制，重新迁移作业。

- 步骤1** 通过作业配置中的源端和目标端连接名称，进入到cdm作业管理—>连接管理，找到该连接器名称。
- 步骤2** 编辑连接器，显示高级属性—>连接属性—>添加：属性名称socketTimeout 值：36000（单位为秒），测试连接，保存。



- 步骤3** 重新启动作业，等待迁移任务执行成功。

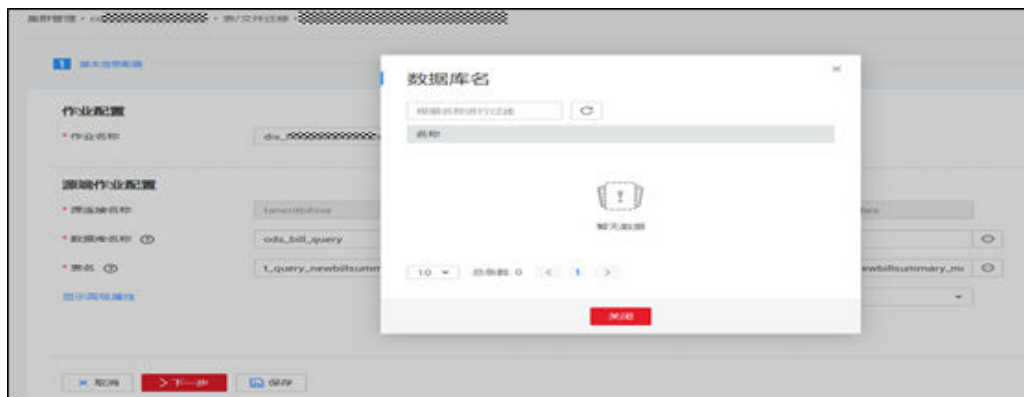
----结束

## 3.31 CDM 集群 Hive 连接无法查询库和表的内容

### 问题描述

cdm集群hive连接无法查询到数据库和表的内容，手动配置库和表后字段可以显示，但报错hive 客户端初始化失败，无效的方案：

get\_table\_req。



## 解决方案

1. 用户的MRS集群是1.8.1，CDM为2.6.0。
2. 报错看CDM封装的Hive SDK无法识别Hive数据源，但Hive连接器测试连通性是正常的，于是仔细检查Hive的连接器配置的参数。

安全集群MRS Manager用户、用户组和角色配置都正确，发现Hive版本配置的为HIVE\_3\_X。

|                             |                         |
|-----------------------------|-------------------------|
| * 名称                        | hive_link               |
| * 连接器                       | Hive                    |
| * Hadoop类型                  | MRS                     |
| * Manager IP <span>?</span> | <input type="text"/> 选择 |
| 认证类型                        | KERBEROS                |
| * Hive版本 <span>?</span>     | HIVE_3_X                |

3. 由于MRS1.8.1集群hive版本为1.2.1，故应该选择hive\_1\_X。正确修改连接器配置，重新创建作业正常。

## 3.32 创建 FusionInsight HDFS 连接报错 get filesystem 怎么解决?

### 问题描述

创建FusionInsight HDFS数据连接时，测试连通性提示获取文件系统失败的问题。



## 解决方案

客户使用的管理ip有误，正确的ip使用的是集群的一个浮动ip，端口使用HDFS的webui的端口即可解决。

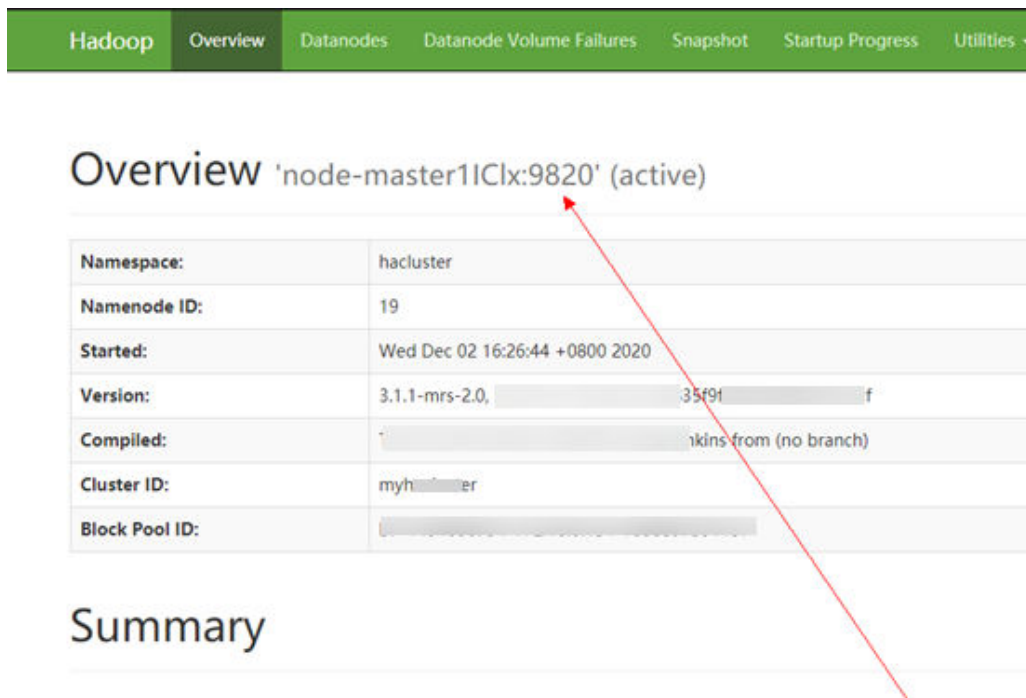
```
[root@node-master1IClx HDFS]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 1[redacted] netmask 255.255.255.0 broadcast 1[redacted]
    ether f[redacted] txqueuelen 1000 (Ethernet)
    RX packets 299702871 bytes 44983338695 (41.8 GiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 251978522 bytes 183960299702 (171.3 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0:wsom: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet [redacted] netmask 255.255.255.0 broadcast [redacted]
    ether [redacted]:3d:5[redacted] txqueuelen 1000 (Ethernet)

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 1[redacted]1 netmask 255.0.0.0
    loop txqueuelen 0 (Local Loopback)
    RX packets 1041006791 bytes 359182733000 (334.5 GiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1041006791 bytes 359182733000 (334.5 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@node-master1IClx HDFS]#
```





### 3.33 Mysql 导入数据到 DLI，快执行完时失败了提示 Invoke DLI service api failed 错误怎么解决？

#### 问题描述

导入了4000W数据，快执行完时报如下错误。



```
com.huawei.dli.restapi.ApiException: Bad Request
    at com.huawei.dli.restapi.ApiClient.handleResponse(ApiClient.java:1073)
    at com.huawei.dli.restapi.ApiClient.execute(ApiClient.java:989)
    at com.huawei.dli.restapi.api.RestApi.executeWithHttpInfo(RestApi.java:4568)
    at com.huawei.dli.restapi.api.RestApi.execute(RestApi.java:4552)
    at com.huawei.dli.sdk.SQLJob.submitSqlNoRetry(SQLJob.java:303)
    at com.huawei.dli.sdk.SQLJob.asyncSubmit(SQLJob.java:127)
    at com.huawei.dli.sdk.UploadJob.beginCommit(UploadJob.java:220)
    at org.apache.sqoop.connector.uquery.intf.impl.UQueryUploadJob.beginCommit(UQueryUploadJob.java:27)
    at org.apache.sqoop.connector.uquery.UQueryLoader.waitForUQueryCompletion(UQueryLoader.java:207)
    at org.apache.sqoop.connector.uquery.UQueryLoader.load(UQueryLoader.java:103)
    at org.apache.sqoop.connector.uquery.UQueryLoader.load(UQueryLoader.java:32)
    at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.runInterval(SqoopOutputFormatLoadExecutor.java:747)
    at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.run(SqoopOutputFormatLoadExecutor.java:629)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
2022-02-28 15:47:13.456|ERROR|LocalJobRunner Map Task #0 #loader|||o.a.s.c.u.intf.impl.UQueryUploadJob DLI ERROR!
```

## 故障分析

**步骤1** 结合报错，考虑是DLI目的端写入问题。但因日志截图不全，进入CDM集群，查看客户作业日志。

```
2022-02-28 16:55:05.650|ERROR|LocalJobRunner Map Task #1 #loader|||o.a.s.connector.uquery.UQueryLoader An error occurred when loading data to DLI.
org.apache.sqoop.common.SqoopException: UQUERY_CONNECTOR_0001:Invoke DLI service api failed, failed reason is %s.
    at org.apache.sqoop.connector.uquery.intf.impl.UQueryUploadJob.beginCommit(UQueryUploadJob.java:30)
    at org.apache.sqoop.connector.uquery.UQueryLoader.waitForUQueryCompletion(UQueryLoader.java:207)
    at org.apache.sqoop.connector.uquery.UQueryLoader.load(UQueryLoader.java:103)
    at org.apache.sqoop.connector.uquery.UQueryLoader.load(UQueryLoader.java:32)
    at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.runInterval(SqoopOutputFormatLoadExecutor.java:747)
    at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.run(SqoopOutputFormatLoadExecutor.java:629)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
Caused by: com.huawei.dli.sdk.exception.DLIException: HTTP Status:400 Bad Request; DLI.0001 queue type must be sql
    at com.huawei.dli.sdk.SQLJob.submitSqlNoRetry(SQLJob.java:311)
    at com.huawei.dli.sdk.SQLJob.asyncSubmit(SQLJob.java:127)
    at com.huawei.dli.sdk.UploadJob.beginCommit(UploadJob.java:220)
    at org.apache.sqoop.connector.uquery.intf.impl.UQueryUploadJob.beginCommit(UQueryUploadJob.java:27)
    at org.apache.sqoop.connector.uquery.UQueryLoader.waitForUQueryCompletion(UQueryLoader.java:207)
    at org.apache.sqoop.connector.uquery.UQueryLoader.load(UQueryLoader.java:103)
    at org.apache.sqoop.connector.uquery.UQueryLoader.load(UQueryLoader.java:32)
    at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.runInterval(SqoopOutputFormatLoadExecutor.java:747)
    at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.run(SqoopOutputFormatLoadExecutor.java:629)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
10 common frames omitted
```

**步骤2** 根据分析步骤一报错，考虑是选错了队列类型，需要选择SQL队列。用户应该是先写到OBS文件，然后通过外表映射导入到DLI表。数据基本已经完成，最终映射时候报错，因为这种场景需要使用DLI的SQL队列。

----结束

## 解决方案

联系用户核实，确实选择队列不是SQL队列。并且查询资源得知，账户名下队列没有SQL队列，让用户购买DLI-SQL队列进行迁移同步。

## 3.34 作业配置添加字段，MongoDB 字段映射存在问题

### 问题描述

CDM作业配置源端MongoDB添加字段，目的端MongoDB数据库字段映射，作业运行后，目的端数据库查看，数据存在问题，没有迁移成功。

| 源字段             |                         |      | 目的字段            |        |    |
|-----------------|-------------------------|------|-----------------|--------|----|
| 名称              | 样值                      | 操作   | 名称              | 类型     | 操作 |
| platform_id     | 7                       | 🔍 🗑️ | platform_id     | string | 🗑️ |
| product_id      |                         | 🔍 🗑️ | product_id      | string | 🗑️ |
| version         |                         | 🔍 🗑️ | version         | string | 🗑️ |
| download_am     | 1                       | 🔍 🗑️ | download_am     | bigint | 🗑️ |
| app_name        |                         | 🔍 🗑️ | app_name        | string | 🗑️ |
| bundle_id       | com.easysay.learningpth | 🔍 🗑️ | bundle_id       | string | 🗑️ |
| create_time     | 2022-01-19 15:03:54.941 | 🔍 🗑️ | create_time     | string | 🗑️ |
| description     | 普通话发音学习测试, 播音主持...      | 🔍 🗑️ | description     | string | 🗑️ |
| enterprise_id   |                         | 🔍 🗑️ | enterprise_id   | string | 🗑️ |
| enterprise_name |                         | 🔍 🗑️ | enterprise_name | string | 🗑️ |
| external_id     | a                       | 🔍 🗑️ | external_id     | string | 🗑️ |
| oss_id          |                         | 🔍 🗑️ | oss_id          | string | 🗑️ |
| update_time     | 2022-01-19 15:03:54.941 | 🔍 🗑️ | update_time     | string | 🗑️ |
| rate            | 1                       | 🔍 🗑️ | rate            | double | 🗑️ |
| app_category    | 1                       | 🔍 🗑️ | app_category    | string | 🗑️ |
| subheading      | 1                       | 🔍 🗑️ | subheading      | string | 🗑️ |

9 select distinct release\_time from bi\_temp.product\_info\_test\_bak\_bak

| compatibility | age_grading | release_time | app_support | platform_id | app_category | app_pic | download_am | external_id | enterprise_na... | app_name |
|---------------|-------------|--------------|-------------|-------------|--------------|---------|-------------|-------------|------------------|----------|
| 1             | 1           | 1            | 1           | 7           | 1            | 1       | 1           |             |                  |          |
| 1             | 1           | 1            | 1           | 7           | 1            | 1       | 1           |             |                  |          |
| 1             | 1           | 1            | 1           | 4           | 1            | 1       | 1           |             |                  |          |
| 1             | 1           | 1            | 1           | 3           | 生活休闲         | 1       | 1           |             |                  |          |
| 1             | 1           | 1            | 1           | 3           | 影音播放         | 1       | 1           |             |                  |          |
| 1             | 1           | 1            | 1           | 4           | 生活           | 1       | 1           |             |                  |          |

## 故障分析

**步骤1** 查看文档提示CDM通过获取样值的方式无法获得所有列。

**步骤2** 添加字段，因为MongoDB是文档数据库，没有scheme概念。CDM字段映射取的是第一条的json key。CDM支持combine()函数，可以把非公共的列封装为一个列。

----结束

## 解决方案

**步骤1** 使用MongoDB Reader插件读出数据时，combine()支持合并MongoDB document中的多个字段为一个JSON串（多个字段合并成一个json串，当做一个字段到目的端）。

| 源字段             |                         |      | 目的字段             |        |    |
|-----------------|-------------------------|------|------------------|--------|----|
| 名称              | 样值                      | 操作   | 名称               | 类型     | 操作 |
| external_id     |                         | 🔍 🗑️ | externalId       | string | 🗑️ |
| rank_year       |                         | 🔍 🗑️ | rank_year        | string | 🗑️ |
| type            |                         | 🔍 🗑️ | type             | int    | 🗑️ |
| _class          |                         | 🔍 🗑️ | owner_enterprise | string | 🗑️ |
| create_time     |                         | 🔍 🗑️ | create_time      | string | 🗑️ |
| enterprise_id   |                         | 🔍 🗑️ | enterprise_id    | string | 🗑️ |
| enterprise_name |                         | 🔍 🗑️ | enterprise_name  | string | 🗑️ |
| is_valid        |                         | 🔍 🗑️ | is_valid         | int    | 🗑️ |
| province_code   |                         | 🔍 🗑️ | province_code    | int    | 🗑️ |
| rank_name       |                         | 🔍 🗑️ | rank_name        | string | 🗑️ |
| release_time    | 2019-12-08 23:46:40.000 | 🔍 🗑️ | release_time     | string | 🗑️ |
| update_time     | 2022-01-19 15:00:28.932 | 🔍 🗑️ | update_time      | string | 🗑️ |
| combine         | combine()               | 🔍 🗑️ | combine          | string | 🗑️ |

**步骤2** 目的端数据库把同步过去数据，通过SQL分解处理。如下图。

SQL脚本

```
1 insert overwrite table bi_temp.enterprise_ranking_test
2 select
3 enterprise_name,
4 external_id,
5 type,
6 rank_name,
7 rank_year,
8 province_code,
9 get_json_object(combine,'$.extra') as extra,
10 is_valid,
11 get_json_object(combine,'$.owner_enterprise') as owner_enterprise,
12 release_time,
13 enterprise_id,
14 create_time,
15 update_time,
16 get_json_object(combine,'$.commend_party') as commend_party,
17 combine
18 from
19 bi_temp.enterprise_ranking_test
20
```

### 📖 说明

这里不影响作业映射已有字段，combine()中是包含所有新增字段的json串，目的端sql进行处理即可获取数据。

----结束

### 3.35 DLI 外表(OBS 文件)迁移 DWS 某字段转义，带有“\”

#### 问题描述

DLI 外表CDM服务将数据迁移到DWS ( GaussDB ) 时候，有个字段迁移后多了一对引号，字段本身的引号多了转义符，其他字段没问题。

源端：

执行成功

执行语句: select \* from iaas\_dm\_obs\_tenant\_view\_info where secure\_score\_factors is not null  
作业ID: b4be761b-cfb7-4f0e-b6a4-903d06b5dab  
查询耗时4.90s, 已扫描52.40 KB, 控制台最多显示1000条数据。

| indard_ia_count | glacier_size | glacier_count | total_size | total_count | tenant_level | illegal_tag | secure_score           | secure_score_fact... | other_indicators | mainland_recommen... | update_time |
|-----------------|--------------|---------------|------------|-------------|--------------|-------------|------------------------|----------------------|------------------|----------------------|-------------|
| 0               | 0            | 7504          | 8          | V5          | null         | 100         | {'base_line_deducti... | {'inbound_atta...    | g                | 1640815349           |             |
| 0               | 0            | 27472480      | 2747248    | V5          | null         | 80          | {'base_line_deducti... | {'inbound_atta...    | g                | 1640815349           |             |
| 0               | 0            | 1249273...    | 4          | V5          | null         | 80          | {'base_line_deducti... | {'inbound_atta...    | g                | 1640815349           |             |
| 0               | 0            | 5753260       | 188        | V0          | null         | 80          | {'base_line_deducti... | {'inbound_atta...    | s                | 1640815349           |             |
| 0               | 0            | 294128        | 92         | V4          | null         | 80          | {'base_line_deducti... | {'inbound_atta...    | s                | 1640815349           |             |
| 0               | 0            | 2420344...    | 2448       | V4          | null         | 80          | {'base_line_deducti... | {'inbound_atta...    | s                | 1640815349           |             |
| 0               | 0            | 1779301...    | 776        | V0          | null         | 60          | {'base_line_deducti... | {'inbound_atta...    | default          | 1640815349           |             |
| 0               | 0            | 5966242...    | 12538588   | V5          | null         | 58          | {'base_line_deducti... | {'inbound_atta...    | g                | 1640815349           |             |
| 0               | 0            | 3064382...    | 24         | V5          | null         | 98          | {'base_line_deducti... | {'inbound_atta...    | g                | 1640815349           |             |
| 0               | 0            | 1325010...    | 16         | V5          | null         | 98          | {'base_line_deducti... | {'inbound_atta...    | g                | 1640815349           |             |

目的端：

```
1 select * from tenant_info where SECURE_SCORE_FACTORS is not null
```

行历史 执行结果 执行计划

信息 结果集1

| IA_COUNT      | TOTAL_SIZE | TOTAL_COUNT | SECURE_SCORE_FACTORS  |
|---------------|------------|-------------|---|
| 0             |            | 0           | "{'base_line_deduction': {'critical': 0, 'low': 0, 'high': 0, 'info': 0, 'mid': 0}, 'three  |
| 12492734464   |            | 4           | "{'base_line_deduction': {'critical': 0, 'low': 0, 'high': 0, 'info': 0, 'mid': 7}, 'three  |
| 2837319312    |            | 96          | "{'base_line_deduction': {'critical': 0, 'low': 0, 'high': 0, 'info': 0, 'mid': 0}, 'three  |
| 1234183651328 |            | 48          | "{'base_line_deduction': {'critical': 0, 'low': 0, 'high': 0, 'info': 0, 'mid': 16}, 'three |

#### 故障分析

1. 根据截图可以看出，源端样值中有符号：{ 括号 ” 引号，等特殊符号，jdbc驱动会字段转义，导致目的端显示带有转义符号。
2. DLI外表及OBS桶存储，及文件到表迁移，可以考虑源端作业配置加上包围符号即可，包围符双引号“”，单个双引号”。

## 解决方案

在OBS作业源端参数配置中，配置开启使用包围符号，单个双引号“”，开启使用包围符号，选择“是”即可。

|      |       |  |
|------|-------|--|
| 高级属性 | 换行符   | 文件中的换行符，默认自动识别“\n”、“\r”或“\r\n”。当“文件格式”选择为“CSV格式”时，才有该参数。 |
|      | 字段分隔符 | 文件中的字段分隔符，使用Tab键作为分隔符请输入“\t”。当“文件格式”选择为“CSV格式”时，才有该参数。   |
|      | 使用包围符 | 选择“是”时，包围符内的字段分隔符会被视为字符串值的一部分，目前CDM默认的包围符为：“”。           |

## 3.36 执行 Postgresql-to-Hive 迁移作业报错 “Error occurs during loader run”

### 问题描述

用户使用CDM服务，从源端pg迁移数据到目的端hive界面报错提示“Error occurs during loader run”。

### 故障分析

1. 排查客户CDM昨天日志报错发现报错：2021-09-29 10:35:32,638 ERROR LocalJobRunner Map Task #13 #loader [org.apache.sqoop.connector.hive.hiveWriter.HiveOrcWriter:83] Create file system error.

```
java.nio.file.AccessDeniedException: obs-itotshujuru-hu-bingxing-fangcongyang: doesBucketExist on obs-itotshujuru-hu-bingxing-fangcongyang: com.obs.services.exception.ObsException: Error message:Request Error.OBS servcie Error Message. -- ResponseCode: 403
```

```
2021-09-29 10:35:32,638 ERROR LocalJobRunner Map Task #13 #loader [org.apache.sqoop.connector.hive.hiveWriter.HiveOrcWriter:83]
Create file system error.
java.nio.file.AccessDeniedException: obs-itotshujuru-hu-bingxing-fangcongyang: doesBucketExist on obs-itotshujuru-hu-bingxing-fangcongyang: com.obs.services.exception.ObsException: Error message:Request Error.OBS servcie Error Message. -- ResponseCode: 403
ResponseStatus: Forbidden, RequestId: xxxxxxxx, HostId: xxxxxxxx
at org.apache.hadoop.fs.obs.OBSUtils.translateException(OBSUtils.java:527)
at org.apache.hadoop.fs.obs.OBSFileSystem.verifyBucketExists(OBSFileSystem.java:326)
at org.apache.hadoop.fs.obs.OBSFileSystem.initialize(OBSFileSystem.java:280)
at org.apache.hadoop.fs.FileSystem.createFileSystem(FileSystem.java:3382)
at org.apache.hadoop.fs.FileSystem.access$200(FileSystem.java:124)
at org.apache.hadoop.fs.FileSystem$Cache.getInternal(FileSystem.java:3431)
at org.apache.hadoop.fs.FileSystem$Cache.get(FileSystem.java:3399)
at org.apache.hadoop.fs.FileSystem.get(FileSystem.java:477)
at org.apache.hadoop.fs.Path.getFileSystem(Path.java:361)
at org.apache.sqoop.connector.hive.hiveWriter.HiveOrcWriter.createWriterOptions(HiveOrcWriter.java:78)
at org.apache.sqoop.connector.hive.hiveWriter.HiveOrcWriter.initialize(HiveOrcWriter.java:66)
at org.apache.sqoop.connector.hive.HiveUtils.getHiveWrite(HiveUtils.java:702)
at org.apache.sqoop.connector.hive.impl.HiveLocalService.lambda$initLoad$15(HiveLocalService.java:564)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:422)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1729)
at org.apache.sqoop.auth.UGIExecutor.execute(UGIExecutor.java:47)
at org.apache.sqoop.auth.KerberosAuthenticator.execute(KerberosAuthenticator.java:63)
at org.apache.sqoop.connector.hive.impl.HiveLocalService.initLoad(HiveLocalService.java:565)
at org.apache.sqoop.connector.hive.HiveLoader.load(HiveLoader.java:58)
at org.apache.sqoop.connector.hive.HiveLoader.load(HiveLoader.java:22)
```

2. 根据报错“hiveWriter.HiveOrcWriter:83] Create file system error”以及“Error.OBS servcie Error Message. -- ResponseCode: 403”考虑是Hive同步表到OBS目录报错。用户配置连接器时候，没有打开OBS开关。
3. 检查连接器配置，发现没有打开开关，参数含义是“是否支持OBS存储，如果Hive表数据存储在OBS，需要打开此开关”。

|              |  |
|--------------|--|
| * 名称         | <input type="text" value="hive"/>  |
| * 连接器        | <input type="text" value="Hive"/>  |
| * Hadoop类型   | <input type="text" value="MRS"/>   |
| * Manager IP | <input type="text" value="[Redacted]"/> <a href="#">选择</a>                       |
| 认证类型         | <input type="text" value="SIMPLE"/>  |
| * Hive版本     | <input type="text" value="HIVE_3_X"/>  |
| * 用户名        | <input type="text" value="[Redacted]"/>  |
| * 密码         | <input type="password"/>   |
| * OBS支持      | <input type="checkbox" value="是"/> <input checked="" type="checkbox" value="否"/> |
| * 运行模式       | <input type="text" value="EMBEDDED"/>  |
| 是否使用集群配置     | <input type="checkbox" value="是"/> <input checked="" type="checkbox" value="否"/> |

## 解决方案

修改连接配置，打开Hive连接中的OBS开关，重新输入密码。

|                           |  |
|---------------------------|--|
| * 名称                      | <input type="text" value="hive"/>                                |
| * 连接器                     | <input type="text" value="Hive"/>                                |
| * Hadoop类型                | <input type="text" value="MRS"/>                                 |
| * Manager IP <sup>?</sup> | <input type="text" value="选择"/>                                  |
| 认证类型                      | <input type="text" value="SIMPLE"/>                              |
| * Hive版本 <sup>?</sup>     | <input type="text" value="HIVE_3_X"/>                            |
| * 用户名 <sup>?</sup>        | <input type="text" value="用户名"/>                                 |
| * 密码                      | <input type="password" value="..."/>                             |
| * OBS支持 <sup>?</sup>      | <input checked="" type="checkbox"/> 是 <input type="checkbox"/> 否 |
| * 访问标识(AK) <sup>?</sup>   | <input type="text" value="AK"/>                                  |
| * 密钥(SK) <sup>?</sup>     | <input type="text" value=".."/>                                  |
| * 运行模式 <sup>?</sup>       | <input type="text" value="EMBEDDED"/>                            |
| 是否使用集群配置 <sup>?</sup>     | <input type="checkbox"/> 是 <input checked="" type="checkbox"/> 否 |

### 3.37 迁移 Mysql 到 DWS 报错 “Lost connection to MySQL server during query” 怎么处理？

#### 问题描述

Mysql-TO-DWS迁移过程中，报错 “GENERIC\_JDBC\_CONNECTOR\_0904:ERROR occurs while retrieving data from result. Cause : closed connection:stream closed con:192.168.XX.XX.”。

```
at java.util.concurrent.FutureTask.run(FutureTask.java:209)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
2021-09-02 09:15:34,834 ERROR LocalJobRunner Map Task Executor #0 [org.apache.sqoop.job.mr.SqoopMapper:181] ERROR occurs during extractor run.
org.apache.sqoop.common.SqoopException: GENERIC_JDBC_CONNECTOR_0904:ERROR occurs while retrieving data from result. Cause : closed connection:stream closed con:192.168.0.32.
at org.apache.sqoop.utils.ERRORResourceUtil.throwSqoopException(ERRORResourceUtil.java:29)
at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:136)
at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:60)
at org.apache.sqoop.job.mr.SqoopMapper.runInternal(SqoopMapper.java:178)
at org.apache.sqoop.job.mr.SqoopMapper.run(SqoopMapper.java:88)
at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:793)
at org.apache.hadoop.mapred.MapTask.run(MapTask.java)
at org.apache.hadoop.mapred.LocalJobRunner$Job$MapTaskRunnable.run(LocalJobRunner.java:270)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
```



## 故障分析

1. 考虑用户数据量大，或连接中断异常导致报错，检查客户连接高级属性配置，是否配置超时时间参数设置及设置情况：

\* 密码 ②

使用本地API ②  是  否

隐藏高级属性

一次请求行数 ②

| 属性名称        | 值       | 操作 |
|-------------|---------|----|
| netWriteTim | 300000  | 删除 |
| connectTim  | 6000000 | 删除 |
| socketTimec | 6000000 | 删除 |

连接属性 ②

添加

引用符号 ②

2. 用户连接参数超时时间“600000”毫秒，已经足够大。进一步排查客户Mysql数据库是否异常，将日志中打印查询源端的sql在Mysql客户端执行，看是否报错以及报错情况：

```
SELECT F_MARKET_ID, F_MARKET_NAME, F_MARKET_TYPE_ID, F_LOWEST_FEE, F_EFFECT_DATE, F_EXPIRE_DATE, F_IS_SMS, F_IS_NOTICE, F_SMS_CONTENT, F_NOTICE_CONTENT, F_STATE, F_OPERATOR, F_OPERATOR_NAME, F_OPERATE_TIME, F_MODIFIER, F_MODIFIER_NAME, F_MODIFIED_TIME, F_EXECUTE_TIME, F_EXECUTE_DAY, F_EXECUTE_STATE, F_NUMBEROFPEOPLE, F_EVERPEOPLENUMBER, F_REWARD_TYPE, F_IS_WECHAT, F_WECHAT_CONTENT, F_SHOP_RULE, F_USE_TERMINAL, F_REDPAPER_WISHING, F_REDPAPER_ACTIVITY_NAME, F_REDPAPER_REMARK, F_REDPAPER_SEND_NAME, F_IS_ANAMARKET, F_COMPAREDAY, F_SHOP_USER_DIVIDE, F_Change_Exp_Day, F_Exp_Date, F_Exp_Multiple, F_NUMBEROFPEOPLE_CURRENT, F_CONSUMPTION_CODITION_RULE, F_EXCLUSIVE_PRIVILEGE, F_USE_TYPE, F_MARKET_TREE_CODE, F_TICKET_RULE_OF_MARKET, F_BIRTH_EXPIRE_DATE, F_BIRTH_EFFECT_DATE, F_BIRTH_DATE, F_BIRTH_TYPE, F_BIRTH_CONDISN, F_MER, F_REDPAPER_WISHING_ACCOUNT, F_REDPAPER_ACTIVITY_NAME_ACCOUNT, F_REDPAPER_REMARK_ACCOUNT, F_REDPAPER_SEND_NAME_ACCOUNT, F_CUS_SUPER, F_CUS_GRADE, F_DATE_TYPE, F_OPERATE_STATE, F_DIRECT_TYPE, F_MAX_AWARD_VALUE FROM activity.t_market_plan_main_info WHERE 1=1
> Lost connection to MySQL server during query
> 时间: 4.952s
```

3. 发现执行查询语句，全表查询，报错“Lost connection to MySQL server during query”，再次尝试执行count语句，查询数据，发现成功。
4. 根据分析，考虑是Mysql配置“max\_allowed\_packet”参数太小导致报错，参考以下链接排查，发现Mysql已经设置最大为1G，无法再增大。
5. 据以上分析，发现未能解决，再次回顾排查过程，发现遗漏一点关键点，客户连接器参数配置，“一次请求行数”配置“3000”，可能会导致某批次查询数据超过1G，故而报错。

## 解决方案

1. 用户修改连接器参数配置，“一次请求行数”修改为“1000”。
2. 用户使用where条件语句，根据时间定期迁移部分数据。

## 3.38 迁移 MySQL 到 DLI 字段类型转换报错 For input string: "false"怎么处理?

### 问题描述

MySQL-TO-DLI迁移报错，java.lang.NumberFormatException: For input string: "false"。

```
at java.lang.Thread.run(Thread.java:748)
2021-07-28 16:20:46,854 ERROR query-loader-1 [org.apache.sqoop.connector.uquery.processor.Dataconsumer:174] failed to convert the record.
java.lang.NumberFormatException: For input string: "false"
at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
at java.lang.Integer.parseInt(Integer.java:580)
at java.lang.Integer.parseInt(Integer.java:615)
at org.apache.sqoop.connector.uquery.processor.Dataconsumer.convert(Dataconsumer.java:271)
at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:172)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)
2021-07-28 16:20:46,854 ERROR query-loader-0 [org.apache.sqoop.connector.uquery.processor.Dataconsumer:174] failed to convert the record.
java.lang.NumberFormatException: For input string: "false"
at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
at java.lang.Integer.parseInt(Integer.java:580)
at java.lang.Integer.parseInt(Integer.java:615)
at org.apache.sqoop.connector.uquery.processor.Dataconsumer.convert(Dataconsumer.java:271)
at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:172)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)
2021-07-28 16:20:46,854 ERROR query-loader-1 [org.apache.sqoop.connector.uquery.processor.Dataconsumer:174] failed to convert the record.
java.lang.NumberFormatException: For input string: "false"
at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
at java.lang.Integer.parseInt(Integer.java:580)
at java.lang.Integer.parseInt(Integer.java:615)
at org.apache.sqoop.connector.uquery.processor.Dataconsumer.convert(Dataconsumer.java:271)
at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:172)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)
2021-07-28 16:20:46,854 ERROR query-loader-0 [org.apache.sqoop.connector.uquery.processor.Dataconsumer:174] failed to convert the record.
java.lang.NumberFormatException: For input string: "false"
at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
at java.lang.Integer.parseInt(Integer.java:580)
at java.lang.Integer.parseInt(Integer.java:615)
at org.apache.sqoop.connector.uquery.processor.Dataconsumer.convert(Dataconsumer.java:271)
at org.apache.sqoop.connector.uquery.processor.Dataconsumer.run(Dataconsumer.java:172)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
```

### 故障分析

1. 根据报错分析，考虑是客户侧字段类型转换存在问题，将值为"false"的bool类型转为int类型报错。进一步排查作业配置第二步，字段映射界面，查看对应关系。

|                      |                    |              |   |                        |        |
|----------------------|--------------------|--------------|---|------------------------|--------|
| udid                 | 9C4D2F84CB4A6A...  | VARCHAR(100) | ↔ | udid                   | string |
| os                   | Android10          | VARCHAR(100) | ↔ | os                     | string |
| device               | HUAWEI ALP-AL00    | VARCHAR(100) | ↔ | device                 | string |
| nation               | CN                 | VARCHAR(100) | ↔ | nation                 | string |
| login_tag            | 1.2.1.3            | VARCHAR(100) | ↔ | login_tag              | string |
| support_gpu_insta... | 0                  | TINYINT      | ↔ | support_gpu_instancing | int    |
| language             | zh-CN              | VARCHAR(100) | ↔ | language               | string |
| runm                 | 5.94 GB            | VARCHAR(100) | ↔ | runm                   | string |
| netcode              | 46002              | VARCHAR(100) | ↔ | netcode                | string |
| nettype              | WIFI               | VARCHAR(100) | ↔ | nettype                | string |
| devres               | Point(1080, 1772)  | VARCHAR(100) | ↔ | devres                 | string |
| cpunum               | 8                  | VARCHAR(100) | ↔ | cpunum                 | string |
| cpumaxf              | 1844000            | VARCHAR(100) | ↔ | cpumaxf                | string |
| cpuminf              | 509000             | VARCHAR(100) | ↔ | cpuminf                | string |
| sdkid                | 1627314838169-7... | VARCHAR(100) | ↔ | sdkid                  | string |
| step_id              | 60001001           | INT          | ↔ | step_id                | int    |

2. 根据上一步字段映射分析，其中"support\_gpu\_instancing"字段源端为TINYINT类型，源端值为"0"或"1"，实际是"false"或"true"。迁移到目的端INT类型的字段中会报错，提示类型转换错误，因为Mysql会自动识别将"0"或"1"转换为"false"或"true"。

## 解决方案

可通过以下两种方式解决问题：

- 修改目的端建表语句，INT类型为BOOL类型；
- 源端修改MySql参数，将该"mysql.bool.type.transform"参数设置为"false"。

其中第二种方式可以在CDM的Mysql连接器高级属性中添加：

连接管理-Mysql连接-高级属性-添加参数 mysql.bool.type.transform 值为false，再次输入密码保存即可。

隐藏高级属性

一次请求行数 ?

一次提交行数 ?

SSL加密 ?  是  否

| 属性名称         | 值                                  | 操作 |
|--------------|------------------------------------|----|
| mysql.bool.t | <input type="text" value="false"/> | 删除 |

连接属性 ?

引用符号 ?

## 3.39 迁移 MySQL 到 DWS，TINYINT 类型迁移报错

### 问题描述

使用CDM从MySQL迁移到DWS，运行作业报错“ERROR:value '-1'is out of range for 8 b-bit integer”。

### 错误详情

作业名称 vmall\_mms\_rt.tbl\_activity\_baseinfo

错误详情

```
org.apache.sqoop.common.SqoopException: GENERIC_JDBC_CONNECTOR_0902:Unable to execute the SQL statement. Cause : ERROR: value "-1" is out of range for 8-bit integer
Where: COPY tbl_activity_baseinfo, line 1, column create_source_type: "-1".
at org.apache.sqoop.utils.ErrorResourceUtil.throwSqoopException(ErrorResourceUtil.java:31)
at org.apache.sqoop.connector.jdbc.Writer.DwsCsvWriter.throwException(DwsCsvWriter.java:444)
at org.apache.sqoop.connector.jdbc.Writer.DwsCsvWriter.endCopy(DwsCsvWriter.java:363)
at org.apache.sqoop.connector.jdbc.Writer.DwsCsvWriter.onClose(DwsCsvWriter.java:308)
at org.apache.sqoop.connector.jdbc.Writer.DwsCsvWriter.close(DwsCsvWriter.java:294)
at org.apache.sqoop.connector.jdbc.GenericJdbcLoader.load(GenericJdbcLoader.java:129)
at org.apache.sqoop.connector.jdbc.GenericJdbcLoader.load(GenericJdbcLoader.java:44)
at
org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.runInterval(SqoopOutputFormatLoadExecutor.java:728)
at org.apache.sqoop.job.mr.SqoopOutput
```

关闭

### 故障分析

1. 根据问题现象，目的端类型不支持值为“-1”插入，检查目的端字段映射，排查映射问题。

|                    |                     |               |  |  |  |                    |               |
|--------------------|---------------------|---------------|--|--|--|--------------------|---------------|
| description        |                     | VARCHAR(2000) |  |  |  | description        | VARCHAR(4000) |
| rule_desc          |                     | VARCHAR(512)  |  |  |  | rule_desc          | VARCHAR(1024) |
| create_by          | admin               | VARCHAR(128)  |  |  |  | create_by          | VARCHAR(256)  |
| create_time        | 2016-07-06 10:16:51 | DATETIME      |  |  |  | create_time        | TIMESTAMP     |
| update_by          | admin               | VARCHAR(128)  |  |  |  | update_by          | VARCHAR(256)  |
| submit_by          |                     | VARCHAR(128)  |  |  |  | submit_by          | VARCHAR(256)  |
| update_time        | 2016-07-06 10:16:51 | DATETIME      |  |  |  | update_time        | TIMESTAMP     |
| multi_lang         |                     | VARCHAR(3000) |  |  |  | multi_lang         | VARCHAR(6000) |
| be_code            | CN                  | VARCHAR(50)   |  |  |  | be_code            | VARCHAR(100)  |
| version            |                     | INT           |  |  |  | version            | INT           |
| extend_type        | 0                   | INT           |  |  |  | extend_type        | INT           |
| extend             | 0                   | VARCHAR(512)  |  |  |  | extend             | VARCHAR(1024) |
| batch_id           | 0                   | BIGINT        |  |  |  | batch_id           | INT           |
| carrier_code       |                     | VARCHAR(50)   |  |  |  | carrier_code       | VARCHAR(100)  |
| create_source_type | -1                  | TINYINT       |  |  |  | create_source_type | INT1          |

2. 根据上一步字段映射情况判断，进一步排查建表语句。

```
1 SET search_path = vsmall_mms_rt;
2 CREATE TABLE tbl_activity_baseinfo (
3     id bigint NOT NULL,
4     code character varying(100),
5     type integer,
6     name character varying(1000),
7     start_time timestamp without time zone,
8     end_time timestamp without time zone,
9     status integer,
10    description character varying(4000),
11    rule_desc character varying(1024),
12    create_by character varying(256),
13    create_time timestamp without time zone,
14    update_by character varying(256),
15    update_time timestamp without time zone,
16    submit_by character varying(256),
17    multi_lang character varying(6000),
18    be_code character varying(100),
19    version integer,
20    extend_type integer,
21    extend character varying(1024),
22    batch_id integer,
23    carrier code character varying(100),
24    create_source_type tinyint
25 )
```

3. 根据以上截图分析，INT1就是DWS字段类型TINYINT的别名，确认字段映射是对的，没有问题。进一步确认DWS TINYINT是否支持范围，为什么报错提示不支持‘-1’的原因，找到DWS字段类型介绍发现DWS TINYINT类型，支持范围为[0,255]，不支持负数，Mysql的TINYINT类型支持范围是[-128,127]。

| 名称             | 描述                           | 存储空间 | 范围   |
|----------------|------------------------------|------|--|
| TINYINT        | 微整数，别名为INT1。                 | 1字节  | 0 ~ 255  |
| SMALLINT       | 小范围整数，别名为INT2。               | 2字节  | -32,768 ~ +32,767                                      |
| INTEGER        | 常用的整数，别名为INT4。               | 4字节  | -2,147,483,648 ~ +2,147,483,647                        |
| BINARY_INTEGER | 常用的整数INTEGER的别名，为兼容Oracle类型。 | 4字节  | -2,147,483,648 ~ +2,147,483,647                        |
| BIGINT         | 大范围的整数，别名为INT8。              | 8字节  | -9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 |

4. SMALLINT支持负数，建议目的端建表使用SMALLINT类型。

## 解决方案

1. 根据问题分析，客户映射字段为INT1就是DWS的TINYINT类型别名，映射是没问题的。
2. 对于DWS来说，TINYINT类型，取值范围是 0 ~ 255，源端是Mysql，有“-1”这种负值，推荐客户使用 SMALLINT（取值范围：-32,768 ~ +32,767）建表。

### 📖 说明

Hive和MySQL的TINYINT类型取值范围都是[-128,127]，而DWS的TINYINT类型取值范围是[0,255]。

## 3.40 数据迁移前后数据量不一致是什么问题？

### 问题描述

使用CDM进行数据迁移，迁移完成后，目标库数据要比原始库多，有的多十几条，有的多几千条。

### 故障分析

根据故障信息分析，考虑是作业配置限制，检查作业配置，发现目的端配置为导入开始前“不清除”，不清除可能存在多次操作，部分数据重复。

## 解决方案

目的端配置为导入开始前“清空全部数据”，验证后，源/目的端条数一致。

## 3.41 创建源数据连接，一直报错用户名和密码错误，但是实际填的没有错

### 问题描述

创建Mysql链接，确认过用户名、密码没有错，同样的配置，在roma上建立数据连接能成功。

### 故障分析

查看后端日志，考虑用户Mysql侧有白名单限制，测试内网相通的另一台ECS Mysql客户端使用这个用户链接。

```
2020-11-26 17:16:43.571 ERROR http-nio-XXXXXXXXXXXXXXXXXXXX-exec-6 [org.apache.sqoop.server.SqoopProtocolServlet:78] Exception in POST http://XXXXXXXXXXXXXXXXXXXX/sqoop/v1/link
org.apache.sqoop.common.SqoopException: CORE_0028:User name or password is incorrect. Detail message : Access denied for user 'XXXXXXXXXXXXXXXXXXXX'(using password: YES)
    at org.apache.sqoop.utils.ErrorResourceUtil.throwSqoopException(ErrorResourceUtil.java:31)
    at org.apache.sqoop.connector.jdbc.GenericJdbcExecutor.throwException(GenericJdbcExecutor.java:1246)
    at org.apache.sqoop.connector.jdbc.GenericJdbcExecutor.getConnection(GenericJdbcExecutor.java:1203)
    at org.apache.sqoop.connector.jdbc.configuration.LinkConfig$ConfigValidator.validate(LinkConfig.java:269)
    at org.apache.sqoop.connector.jdbc.configuration.LinkConfig$ConfigValidator.validate(LinkConfig.java:204)
    at org.apache.sqoop.validation.ConfigValidationRunner.executeValidator(ConfigValidationRunner.java:182)
    at org.apache.sqoop.validation.ConfigValidationRunner.validateArray(ConfigValidationRunner.java:148)
    at org.apache.sqoop.validation.ConfigValidationRunner.validateConfig(ConfigValidationRunner.java:129)
    at org.apache.sqoop.validation.ConfigValidationRunner.validate(ConfigValidationRunner.java:90)
    at org.apache.sqoop.model.ConfigUtils.validateConfigs(ConfigUtils.java:298)
```

## 解决方案

再次核实账号密码无误，是用户使用Mysql账号没有给CDM集群授权，用户侧Mysql数据库需要对cdm集群的ip授权。

## 3.42 数据库写入 OBS 场景，表中小驼峰命名字段，提示字段不存在

### 问题描述

数据库写入OBS场景，表中小驼峰命名字段，提示字段不存在。

### 故障分析

查看日志报PG数据库表字段找不到所致，分析是字段命名使用小驼峰，而PG数据库区分大小写所以无法找到。

### 解决方案

让客户在连接配置高级属性添加包围符配置，问题解决。

## 3.43 CSV 数据类型插入 MySQL 报错 invalid utf-8 charater string "

### 问题描述

迁移作业执行失败，提示invalid utf-8 charater string "。

### 故障分析

考虑是数据格式问题，后端进一步分析日志确认。

### 解决方案

1. 后台排查sqoop日志，考虑源端数据类型格式问题导致异常。
2. 分析源端数据类型，发现数据类型中有脏数据，源端数据类型有问题。
3. 客户CDM界面配置脏数据功能，作业重跑成功，OBS桶排查脏数据类型存在问题，格式不匹配。

## 3.44 定时任务失败，检查连接器连接存在问题

### 问题描述

CDM任务检查网络连通性，源端数据库连接问题，测试连通性提示如下问题：

“请检查IP、主机名、端口填写是否正确，检查网络安全组和防火墙配置是否正确，参考数据库返回消息进行定位。”



## 故障分析

查询集群信息，获取公网IP，从CDM集群curl源端数据库的地址，如下所示。

```
[root@localhost Mike]# curl -vvv jira.huatec.com:8306
* About to connect() to jira.huatec.com port 8306 (#0)
* Trying [REDACTED]...
* Network is unreachable
* Failed connect to jira.huatec.com:8306; Network is unreachable
* Closing connection 0
curl: (7) Failed connect to jira.huatec.com:8306; Network is unreachable
[root@localhost Mike]#
[root@localhost Mike]#
[root@localhost Mike]#
[root@localhost Mike]# ping jira.huatec.com
PING jira.huatec.com ([REDACTED]) 56(84) bytes of data.
From 192.168.1.1 icmp_seq=1 Destination Net Unreachable
From 192.168.1.1 icmp_seq=2 Destination Net Unreachable
From 192.168.1.1 icmp_seq=3 Destination Net Unreachable
From 192.168.1.1 icmp_seq=4 Destination Net Unreachable
^C
--- jira.huatec.com ping statistics ---
6 packets transmitted, 0 received, +4 errors, 100% packet loss, time 5006ms
```

从结果看考虑是CDM集群自身问题。通过EIP查询对应公网IP的绑定情况是未绑定的。

| 弹性IP      | 状态  | IPv4/IPv6 | 已绑定私有IP | 带宽大小                    | 类型         |
|-----------|-----|-----------|---------|-------------------------|------------|
| 49.466.70 | 未绑定 | 4         | --      | 300Mbit/s               | 动态BGP (5_b |
| ID        | 6   | 流量详情      | 带宽名称    |                         |            |
| 已绑定私有IP   | --  |           | 带宽大小    |                         |            |
| 绑定设备名称    | --  |           | 带宽ID    | 98                      | 流量详情       |
| 绑定设备ID    | --  |           | 创建时间    | 2024-03-18 17:53:17.497 |            |

建议先解除绑定，再绑定后问题解决。

## 解决方案

释放掉EIP之后，给CDM集群重新绑定EIP；或者给CDM VPC的委托，然后可以检测这个EIP是否异常。

### 3.45 脏数据导致 CSV 数据类型问题插入 MySQL 报错

#### 问题描述

客户作业失败，提示invalid utf-8 charater string "

```
2020-03-18 17:53:17.497 ERROR OutputFormatLoader-consumer-0 [org.apache.sqoop.connector.jdbc.GenericJdbcExecutor:1073] Caught SQLException
java.sql.SQLException: Invalid utf8 character string: ''
    at com.mysql.jdbc.SQLException.createSQLException(SQLException.java:964)
    at com.mysql.jdbc.MySQLIO.checkERRORPacket(MySQLIO.java:3973)
    at com.mysql.jdbc.MySQLIO.checkERRORPacket(MySQLIO.java:3909)
    at com.mysql.jdbc.MySQLIO.checkERRORPacket(MySQLIO.java:873)
    at com.mysql.jdbc.MySQLIO.sendFileToServer(MySQLIO.java:3876)
    at com.mysql.jdbc.MySQLIO.readResultsForQueryOrUpdate(MySQLIO.java:3110)
    at com.mysql.jdbc.MySQLIO.readAllResults(MySQLIO.java:2341)
    at com.mysql.jdbc.MySQLIO.sqlQueryDirect(MySQLIO.java:2736)
    at com.mysql.jdbc.ConnectionImpl.execSQL(ConnectionImpl.java:2483)
    at com.mysql.jdbc.StatementImpl.executeUpdateInternal(StatementImpl.java:1552)
    at com.mysql.jdbc.StatementImpl.executeLargeUpdate(StatementImpl.java:2607)
    at com.mysql.jdbc.StatementImpl.executeUpdate(StatementImpl.java:1480)
    at org.apache.sqoop.connector.jdbc.GenericJdbcExecutor.executeUpdate(GenericJdbcExecutor.java:465)
    at org.apache.sqoop.connector.jdbc.Writer.MySQLCsvWriter.flush(MySQLCsvWriter.java:203)
    at org.apache.sqoop.connector.jdbc.Writer.MySQLCsvWriter.write(MySQLCsvWriter.java:115)
    at org.apache.sqoop.connector.jdbc.GenericJdbcLoader.load(GenericJdbcLoader.java:77)
    at org.apache.sqoop.connector.jdbc.GenericJdbcLoader.load(GenericJdbcLoader.java:37)
    at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.runInterval(SqoopOutputFormatLoadExecutor.java:687)
    at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutor$ConsumerThread.run(SqoopOutputFormatLoadExecutor.java:562)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
    at java.lang.Thread.run(Thread.java:745)
```

## 故障分析

根据报错，考虑用户数据中存在脏数据，个别字段类型不匹配导致。

## 解决方案

1. 后台排查客户sqoop日志，考虑客户源端数据类型格式问题导致异常（或让客户提供作业日志，或客户界面导出全量日志）。

```
2020-03-18 17:53:17.676 INFO [org.apache.hadoop.mapred.MapTask.closeQuietly(MapTask.java:2064)] Ignoring exception during close for org.apache.hadoop.mapred.MapTaskNewDirectOutputCollector@7b1545bb
org.apache.sqoop.common.SqoopException: MAPRED_EXEC_0234:Error occurs during loader run. Cause : GENERIC JDBC_CONNECTOR_0902:Unable to execute the SQL statement. Cause : invalid utf8 character string: '
    at org.apache.sqoop.uti ls.ErrorResourceUtil.throwSqoopException(ErrorResourceUtil.java:52)
    at org.apache.sqoop.uti ls.ErrorResourceUtil.throwSqoopException(ErrorResourceUtil.java:48)
    at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutorsConsumerThread.runInternal(SqoopOutputFormatLoadExecutor.java:710)
    at org.apache.sqoop.job.mr.SqoopOutputFormatLoadExecutorsConsumerThread.run(SqoopOutputFormatLoadExecutor.java:562)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
    at java.util.concurrent.ThreadPoolExecutor.run(ThreadPoolExecutor.java:617)
    at java.lang.Thread.run(Thread.java:745)
```

2. 分析源端数据类型，发现源端数据类型中有脏数据，源端数据类型有问题。
3. 在CDM作业中配置脏数据功能，作业重跑成功，OBS桶排查脏数据类型存在问题，格式不匹配。

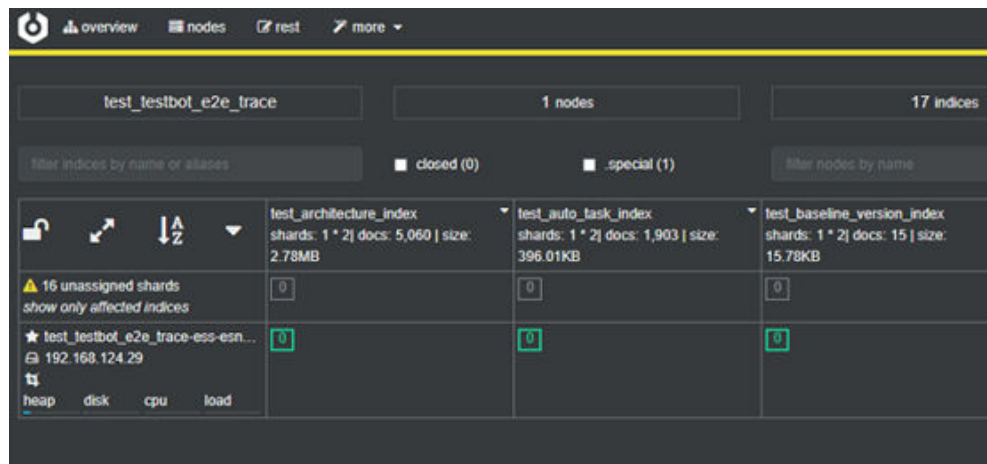
## 3.46 写 ES 报 timeout waiting for connection from pool 错误怎么解决？

### 问题描述

写ES报timeout waiting for connection from pool，且日志中输出多个es\_rejected\_execution\_exception。

### 故障分析

从cerebro界面看到索引只有一个分片。但新建一个索引设成3个分片也是一样会报es\_rejected\_execution\_exception。



继续定位发现记录几乎都写入到了一个分片中。至此问题清楚。是因为产生了热点。

### 解决方案

用户在迁移时有选择主键，也就是用它来替代\_id。计算出来的shard属同一个。

1. 建议用户不选主键，让es自动生成\_id，这样获的hash值比较分散。

2. 如果用户的应用必须用自有主键替代\_id, 则只能建议用性能更好的ES集群。

## 3.47 Oracle 迁移到 DWS 报错 ORA-01555

### 问题描述

Oracle迁移到DWS报错ORA-01555。

```
665 2020-09-21 22:51:02,991 ERROR LocalJobRunner Map Task #3 [org.apache.sqoop.common.SqoopException:111] SqoopException
666 java.sql.SQLException: ORA-01555: snapshot too old: rollback segment number 3 with name "_SYSMU3_2097677531$" too small
667
668 at oracle.jdbc.driver.T4CTTIOer11.processERROR(T4CTTIOer11.java:494)
669 at oracle.jdbc.driver.T4CTTIOer11.processERROR(T4CTTIOer11.java:446)
670 at oracle.jdbc.driver.T4C8Oall.processERROR(T4C8Oall.java:1054)
671 at oracle.jdbc.driver.T4CTTIfun.receive(T4CTTIfun.java:623)
672 at oracle.jdbc.driver.T4CTTIfun.doRPC(T4CTTIfun.java:252)
673 at oracle.jdbc.driver.T4C8Oall.doOALL(T4C8Oall.java:612)
674 at oracle.jdbc.driver.T4CPreparedStatement.doOall8(T4CPreparedStatement.java:226)
675 at oracle.jdbc.driver.T4CPreparedStatement.fetch(T4CPreparedStatement.java:1023)
676 at oracle.jdbc.driver.OracleStatement.fetchMoreRows(OracleStatement.java:3353)
677 at oracle.jdbc.driver.InsensitiveScrollableResultSet.fetchMoreRows(InsensitiveScrollableResultSet.java:736)
678 at oracle.jdbc.driver.InsensitiveScrollableResultSet.absoluteInternal(InsensitiveScrollableResultSet.java:692)
679 at oracle.jdbc.driver.InsensitiveScrollableResultSet.next(InsensitiveScrollableResultSet.java:406)
680 at org.apache.sqoop.connector.jdbc.sql.impl.WrapResultSet.next(WrapResultSet.java:36)
681 at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extractObjectRecord(GenericJdbcExtractor.java:151)
682 at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:129)
683 at org.apache.sqoop.connector.jdbc.GenericJdbcExtractor.extract(GenericJdbcExtractor.java:59)
684 at org.apache.sqoop.job.mr.SqoopMapper.runInternal(SqoopMapper.java:184)
685 at org.apache.sqoop.job.mr.SqoopMapper.run(SqoopMapper.java:81)
686 at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:799)
687 at org.apache.hadoop.mapred.MapTask.run(MapTask.java)
688 at org.apache.hadoop.mapred.LocalJobRunner$JobMapTaskRunnable.run(LocalJobRunner.java:271)
689 at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
690 at java.util.concurrent.FutureTask.run(FutureTask.java:266)
691 at org.apache.sqoop.submission.mapreduce.MapperExecutorGroup$1.lambda$execute$0(MapperExecutorGroup.java:222)
692 at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
693 at java.util.concurrent.FutureTask.run(FutureTask.java:266)
694 at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
695 at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
696 at java.lang.Thread.run(Thread.java:748)
697 Caused by: oracle.jdbc.OracleDatabaseException: ORA-01555: snapshot too old: rollback segment number 3 with name "_SYSMU3_2097677531$" too small
698
699 at oracle.jdbc.driver.T4CTTIOer11.processERROR(T4CTTIOer11.java:498)
700 ... 28 common frames omitted
```

### 故障分析

1. 整库迁移每个表中数据平均2~5亿条。源端10分钟会更新一次数据。
2. CDM不支持实时迁移，但是支持定时迁移，用户10分钟就会有批量数据更新，考虑是迁移任务没有完成，源库已经更新，回滚超时。
3. Oracle报错ORA-01555，数据迁移如果做整表查询，并且查询时间较长时，这个过程有其他用户进行频繁commit操作，Oracle的RBS还比较小，就有可能出现这个问题，详细分析可以参考帖子：[https://blog.csdn.net/SongYang\\_Oracle/article/details/6432182](https://blog.csdn.net/SongYang_Oracle/article/details/6432182)。

### 解决方案

三种解决办法：

- 调小每次查询的数据量。
- 调大Oracle数据的RBS，需要修改数据库配置。
- 减少频繁的commit操作，这个需要调整生产业务逻辑，基本不可能。

## 3.48 FTP 测试连通性失败，报服务器内部错误怎么解决？

### 问题描述

ECS搭建FTP已经尝试root、FTPadmin用户在本地都可以正常登录，安全组21、20端口正常放行，但是在创建FTP的数据连接报服务器内部错误。



## 解决方案

查询后台日志，报错为连接超时，截图如下，可能是安全组限制导致。安全组全部放行进行验证，连通性测试成功。

```
at com.cmc.tomcat.common.RequestHeaderNameValue.invoke(RequestHeaderNameValue.java:68)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:343)
at org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:1616)
at org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:65)
at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:831)
at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1634)
at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
at java.lang.Thread.run(Thread.java:748)
2022-07-25 22:58:05.713|ERROR|[https-1jsse-mio2-172.22.61.85-21203-exec-6][o.a.s.server.SqoopProtocolServlet:66]|Exception in POST https://180.78.233.110:21203/sqoop/v1/Link
org.apache.sqoop.common.SqoopException: FTP_CONNECTOR_0680:Can't connect to the FTP server. Cause: Connection timed out (Connection timed out)
at org.apache.sqoop.utils.ErrorResourceUtil.throwSqoopException(ErrorResourceUtil.java:31)
at org.apache.sqoop.connector.FtpClient.FtpConnectorClient.connect(FtpConnectorClient.java:172)
at org.apache.sqoop.connector.FtpConfiguration.LinkConfig$ConfigValidator.validate(LinkConfig.java:55)
at org.apache.sqoop.connector.FtpConfiguration.LinkConfig$ConfigValidator.validate(LinkConfig.java:58)
at org.apache.sqoop.validation.ConfigValidationRunner.executeValidator(ConfigValidationRunner.java:183)
at org.apache.sqoop.validation.ConfigValidationRunner.validateArray(ConfigValidationRunner.java:149)
at org.apache.sqoop.validation.ConfigValidationRunner.validateConfig(ConfigValidationRunner.java:130)
at org.apache.sqoop.validation.ConfigValidationRunner.validate(ConfigValidationRunner.java:91)
at org.apache.sqoop.model.ConfigUtils.validateConfigs(ConfigUtils.java:318)
at org.apache.sqoop.handler.LinkRequestHandler.createUpdateLink(LinkRequestHandler.java:308)
at org.apache.sqoop.handler.LinkRequestHandler.handleEvent(LinkRequestHandler.java:104)
at org.apache.sqoop.server.v1.LinkServlet.handlePostRequest(LinkServlet.java:71)
at org.apache.sqoop.server.SqoopProtocolServlet.doPost(SqoopProtocolServlet.java:61)
```

针对FTP服务器的防火墙来说，必须允许以下通讯才能支持主动方式FTP：

1. 任何大于1024的端口到FTP服务器的21端口（客户端初始化的连接）。
2. FTP服务器的21端口到大于1024的端口（服务器响应客户端的控制端口）。
3. FTP服务器的20端口到大于1024的端口（服务器端初始化数据连接到客户端的数据端口）。
4. 大于1024端口到FTP服务器的20端口（客户端发送ACK响应到服务器的数据端口）。

## 3.49 CDM 连接 RDS-Mysql ， 除 root 用户外， 其他用户都报错， 怎么办？

### 问题描述

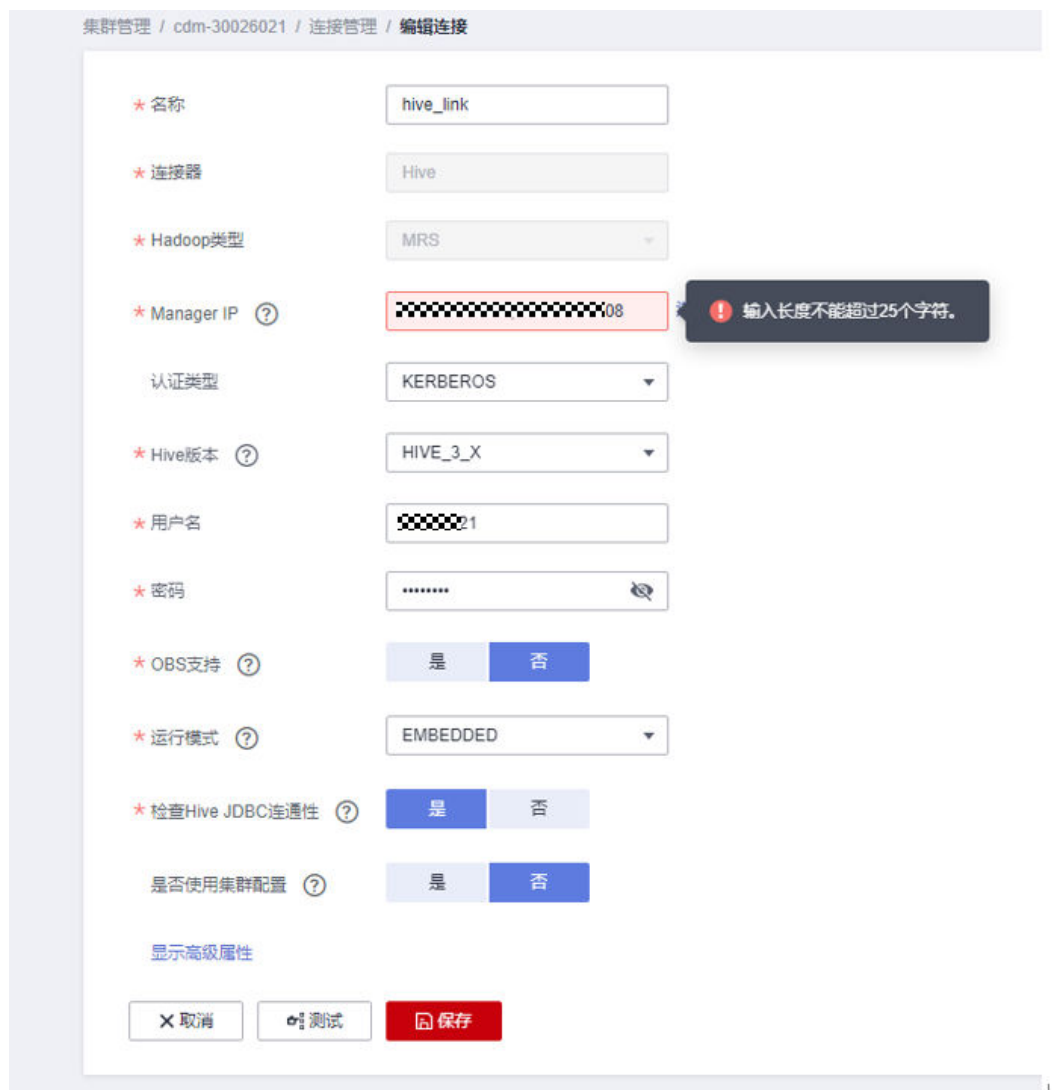
CDM连接RDS-Mysql，除root用户外，其他用户都报错。

## 解决方案

1. 登录服务器，运行命令进入数据库：`mysql -u root -p`，然后输入密码。
2. `mysql>use mysql;`
3. 授权：  
例如想root使用123456从任何主机连接到mysql服务器：  
`mysql>GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;`  
如果想允许用户abc从ip为10.10.50.127的主机连接到mysql服务器,并使用654321作为密码：  
`mysql>GRANT ALL PRIVILEGES ON *.* TO 'abc'@'10.10.50.127' IDENTIFIED BY '654321' WITH GRANT OPTION;`
4. 刷新权限即可：`mysql>FLUSH PRIVILEGES;`

### 3.50 MRS Hive, MRS Kafka, MRS Hudi 数据源创建连接时 IP 长度校验不通过，如何处理？

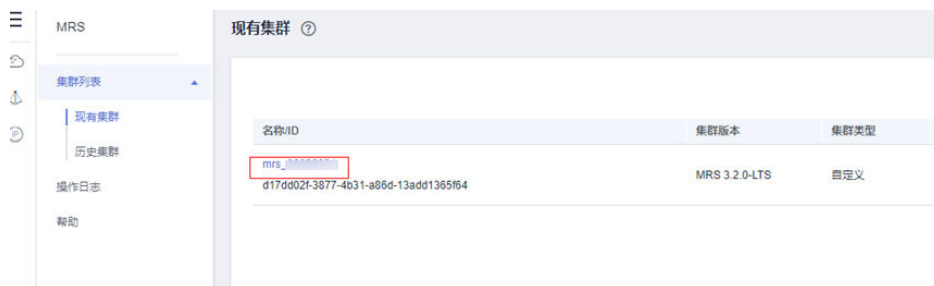
MRS Hive, MRS Kafka, MRS Hudi数据源在创建连接时，单击“选择”选定MRS集群，Manager IP获取到IP如果长度校验不通过，如下图所示。

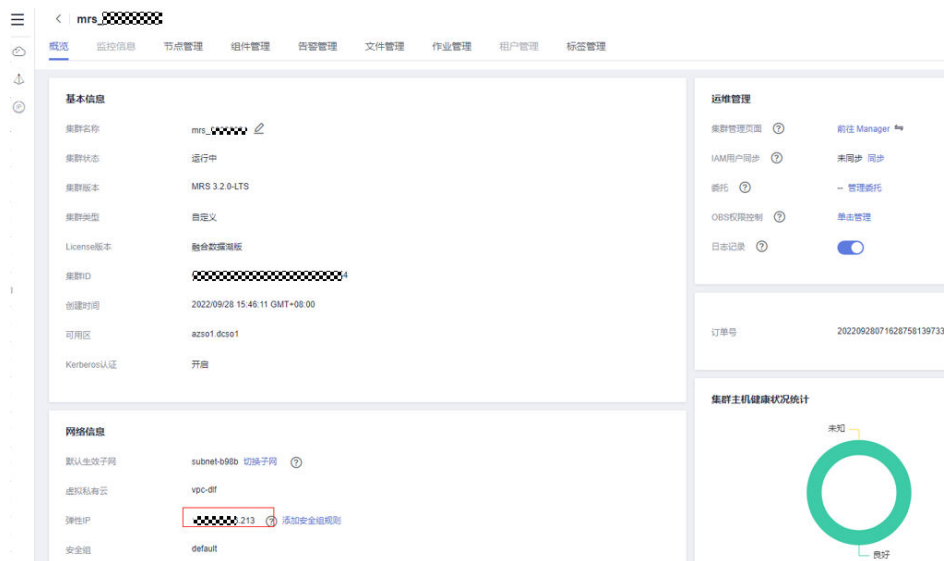


## 解决方法

**步骤1** 单击服务列表，搜索“MRS”，进入MRS控制台。

**步骤2** 单击集群名称进入“现有集群”页面，查询弹性ip，如下图所示。





**步骤3** 通过弹性IP地址，登录MRS集群后台。

用户名: admin

密码: 创建集群时的密码

**步骤4** 执行`ip a | grep wsom`命令查询MRS集群的浮动ip (wsom标志)，如下图所示。

```
[root@node-master1PKcR ~]# ip a | grep wsom
inet 192.168.0.217/24 brd 192.168.0.255 scope global secondary enp4s3:wsom
[root@node-master1PKcR ~]#
```

**步骤5** 返回创建MRS连接界面，输入查询到的IP，创建连接成功。

#### 📖 说明

创建MRS数据源连接时，认证类型要与MRS集群是否开启KERBEROS保持一致。

----结束

## 3.51 CDM 转换空字符串表达式 `StringUtils.defaultIfEmpty` 不生效怎么解决?

### 问题描述

在cdm中使用`StringUtils.defaultIfEmpty(value,"0000-00-00 00:00:00")`表达式期望将空字符串转换为时间格式，但是没有生效。

### 解决方案

客户侧的null是null字符串，而非null空值，所以null字符串无法转化，null空值的查询条件是 `is null`。

## 3.52 当 Hudi 表属性和 hoodie.properties 文件中分区信息不一致时，会导致数据写入失败怎么解决？

### 问题描述

当Hudi表属性和hoodie.properties文件中分区信息不一致时，会导致数据写入失败。

### 解决方案

**步骤1** 执行如下spark-sql语句修改hoodie.properties文件信息。

```
call update_hoodie_properties(table=>'$tableName',  
property=>'hoodie.table.partition.fields', value=>'name')
```

**步骤2** 执行如下spark-sql语句修改表属性信息。

```
alter table $tableName set  
tblproperties('hoodie.datasource.write.partitionpath.field' = '$partitionName1,  
$partitionName12');
```

----结束

## 3.53 当 MySQL、Oracle、PostgreSQL 作为源端时，如果作业报错“Read timed out”怎么处理？

### 问题描述

当MySQL、Oracle、PostgreSQL作为源端时，如果作业报错“Read timed out”怎么处理？

### 解决方案

当MySQL、Oracle、PostgreSQL作为源端时，如果作业报错“Read timed out”，可能是由于源端发生主备倒换，源端出现网络波动等原因导致，建议作业配置失败后重试。

## 3.54 RDS-Mysql 和开源 Mysql 对于 date 类型的范围不一致，可能导致 date 数据转换异常

### 问题描述

使用开源Mysql驱动连接RDS-Mysql，date范围小于1000-01-0的情况下，date数据转换异常。

图 3-18 源端 data 的数据

|   | date1      | datetime1           |
|---|------------|---------------------|
| 1 | 0000-00-00 | 0001-01-01 00:00:00 |



图 3-19 目的端 data 的数据

|   | date1  | datetime1           |
|---|--------|---------------------|
| 1 | (Null) | 0001-01-01 00:00:00 |
| 2 | (Null) | 0001-01-01 00:00:00 |
| 3 | (Null) | 0001-01-01 00:00:00 |

### 故障分析

RDS-Mysql和开源Mysql对于date类型的范围不一致。

开源Mysql的date范围一般为1000-01-0到9999-12-21，而RDS-Mysql支持date范围小于1000-01-0。使用开源Mysql驱动连接RDS-Mysql，在date范围小于1000-01-0的情况下，可能导致date数据转换异常。

### 解决方案

- 不使用该date数值的写入。
- 手动补充该date数值。

## 3.55 通过 json 方式导入的作业，sql server 迁移到 hive，执行作业报错

### 问题描述

客户通过json的方式导入作业，从sql server迁移数据到hive，执行作业报：操作数数据类型bit 对min 运算无效。

图 3-20 报错信息

```
2023-12-27 09:29:54.790 [INFO] [cdm-job-submit-pool11][o.a.s.c.]GenericJdbcFromInitializer:431|Use the partition column: IsDef
2023-12-27 09:29:54.790 [INFO] [cdm-job-submit-pool11][o.a.s.c.]GenericJdbcFromInitializer:172|Using min/max query SELECT MIN(IsDef), MAX(IsDef) FROM "dbo"."BAS_TitleInfoManager" WHERE 1 = 1
com.microsoft.sqlserver.jdbc.SQLServerException: 操作数数据类型 bit 对于 min 运算无效。
    at com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:217)
    at com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1635)
    at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:426)
    at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLServerPreparedStatement.java:372)
    at com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:1794)
    at com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:154)
    at com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:159)
    at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeQuery(SQLServerPreparedStatement.java:284)
    at org.apache.sqoop.connector.jdbc.sql.impl.WrapPreparedStatement.lambda$executeQuery$0(WrapPreparedStatement.java:30)
    at java.security.AccessController.doPrivileged(Native Method)
    at org.apache.sqoop.util.JdbcSanbox.doPrivileged(JdbcSanbox.java:53)
    at org.apache.sqoop.connector.jdbc.sql.impl.WrapPreparedStatement.executeQuery(WrapPreparedStatement.java:30)
    at org.apache.sqoop.connector.jdbc.retrieval.MinMaxValueJointRetrieval.retrieveMinMaxValue(MinMaxValueJointRetrieval.java:198)
    at org.apache.sqoop.connector.jdbc.retrieval.MinMaxValueJointRetrieval.retrieveMinMaxValue(MinMaxValueJointRetrieval.java:122)
    at org.apache.sqoop.connector.jdbc.GenericJdbcFromInitializer.configurePartitionProperties(GenericJdbcFromInitializer.java:515)
    at org.apache.sqoop.connector.jdbc.GenericJdbcFromInitializer.initialize(GenericJdbcFromInitializer.java:166)
    at org.apache.sqoop.connector.jdbc.GenericJdbcFromInitializer.initialize(GenericJdbcFromInitializer.java:160)
    at org.apache.sqoop.driver.job.JobInitiator.initializeConnector(JobInitiator.java:608)
    at org.apache.sqoop.driver.job.JobInitiator.createJobRequest(JobInitiator.java:341)
    at org.apache.sqoop.driver.job.JobSubmissionContext.submitInternal(JobSubmissionContext.java:234)
    at org.apache.sqoop.driver.job.JobSubmissionContext.submit(JobSubmissionContext.java:182)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
2023-12-27 09:29:54.792 [ERROR] [cdm-job-submit-pool11][o.apache.sqoop.common.SqoopException:118]|SqoopException
com.microsoft.sqlserver.jdbc.SQLServerException: 操作数数据类型 bit 对于 min 运算无效。
    at com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:217)
    at com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1635)
    at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:426)
    at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLServerPreparedStatement.java:372)
    at com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:1794)
    at com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:154)
    at com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:159)
    at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeQuery(SQLServerPreparedStatement.java:284)
    at org.apache.sqoop.connector.jdbc.sql.impl.WrapPreparedStatement.lambda$executeQuery$0(WrapPreparedStatement.java:30)
    at java.security.AccessController.doPrivileged(Native Method)
    at org.apache.sqoop.util.JdbcSanbox.doPrivileged(JdbcSanbox.java:53)
    at org.apache.sqoop.connector.jdbc.sql.impl.WrapPreparedStatement.executeQuery(WrapPreparedStatement.java:30)
    at org.apache.sqoop.connector.jdbc.retrieval.MinMaxValueJointRetrieval.retrieveMinMaxValue(MinMaxValueJointRetrieval.java:198)
```

### 故障分析

客户作业中配置的并发数大于1，CDM会选择分区列进行分区迁移，默认情况：取主键字段，如果没有主键字段取的是从元数据里面的第一个字段作为分区字段。

由于客户作业中设置的并发数是8，程序选择IsDef作为分区字段，但是IsDef的类型是bit，不支持min运算所以会报错。

## 解决方案

- 通过修改作业json的方式将作业并发数设置为1即可，参数：“throttlingConfig.numExtractors”。
- 在作业配置界面，源端参数配置的高级属性中，给“抽取分区字段”配置项指定字段即可（建议为源端表的主键或者递增的字段）。

## 3.56 Hudi 源端的作业长时间处于 BOOTING 状态怎么解决？

### 问题描述

在Hudi源端读Hudi作业长时间处于BOOTING状态。

### 原因分析

- 除去Yarn队列资源问题，一般作业是卡在执行Spark SQL读Hudi写Hive临时表，这步执行的速度取决于Hudi表的数据量与Yarn队列剩余资源。
- 作业配置了导入前清空数据，dws表存量数据多，卡在truncate table操作步骤中，默认5分钟超时。

### 解决方案

- 查看Yarn任务，搜索Spark JDBC Server的Yarn任务，找到自己队列下Running Container大于1的任务，查看ApplicationMaster，单击SQL页签，可以看到正在执行的SQL，单击Stages页签，可以看到每条SQL的执行进度。

The screenshot shows the Spark SQL interface. At the top, there is a table with columns: ID, User, Name, Application Type, Queue, Application Priority, StartTime, FinishTime, State, FinalStatus, Running Containers, and Allocated CPU VCores. A row is highlighted with a red box around the 'Queue' column containing 's000\_sbi\_ar' and the 'Running Containers' column containing '4'. Below this, there is a section for 'SQL' with 'Completed Queries: 16'. A table lists completed queries with columns: ID, Description, Submitted, Duration, and Job IDs. Two queries are visible: ID 46 with a duration of 41 s and ID 45 with a duration of 29 ms.

### 说明

CDM在作业BOOTING阶段无法查看日志，如果找不到Yarn任务，请联系CDM运维查看后台日志，获取Application ID。日志形如：

```
2022-06-27 12:18:13.070|INFO |cdm-job-submit-pool7|[o.a.c.f.imps.CuratorFrameworkImpl:356]|Default schema
2022-06-27 12:18:13.189|INFO |cdm-job-submit-pool7|[org.apache.zookeeper.ZooKeeper:1457]|ConnectionId: 0x2902bc5146c6ce25 closed
2022-06-27 12:18:13.190|INFO |cdm-job-submit-pool7|[org.apache.hive.jdbc.HiveConnection:1014]|Login timeout is 120000
2022-06-27 12:18:13.200|INFO |cdm-job-submit-pool7|[org.apache.hive.jdbc.HiveConnection:377]|user login success.
2022-06-27 12:18:13.205|INFO |cdm-job-submit-pool7|[org.apache.hive.jdbc.HiveConnection:476]|will try to open client transport with JDBC Uri: jdbc:hive2://node-
master5vepu:22550/?principal=spark2x/hadoop.4a3bd513_8709_4870_8a2b_751ad7a6667.com@4a380513_8709_4870_8a2b_751ADC7A6667.COM;saslQop=auth-conf;sasl.qop=auth-
conf;serviceDiscoveryNode=zooKeeper;auth=KERBEROS;socketTimeout=120;zooKeeperNamespace=sparkthriftserver2x;user.principal=s999_sbi_otc;user.keytab=/rds/cdm/tomcat/we
bapps/sqoop/HEB-INF/classes/mrs/cdm_s999_sbi_otc_hudi/user.keytab
2022-06-27 12:18:14.086|INFO |cdm-job-submit-pool7|[o.a.s.c.spark.util.SparkJdbcUtils:86]|Connect to: Spark SQL(3.1.1-hw-ei-312005), Yarn Application Id =
application_1654747385752_262875, Driver: Hive JDBC(3.1.0-hw-ei-312005)
2022-06-27 12:18:14.086|INFO |cdm-job-submit-pool7|[o.a.s.c.spark.util.SparkJdbcUtils:136]|execute_pre_sql: set spark.sql.parquet.writeLegacyFormat=true
2022-06-27 12:18:24.610|INFO |log-thread|[o.a.s.c.spark.util.SparkJdbcUtils:182]|INFO | Execution ID: 8916
```

- 联系CDM运维查看后台日志。

## 3.57 Hudi 源端作业，读 Hudi 作业字段映射多了一列 col，作业执行失败怎么处理？

### 问题描述

读Hudi作业字段映射多了一列col，作业执行失败。

```
2022-08-27 21:55:18.781[ERROR][com-joo-submit-pool18][o.apache.spark.common.SparkException:118][SparkException]
java.sql.SQLException: org.apache.hive.service.cli.MysqlException: ERROR running query: org.apache.spark.sql.AnalysisException: cannot resolve ''col'' given input
columns: [spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_hoodie_commit_seqno, spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_hoodie_commit_time,
spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_hoodie_event_time, spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_hoodie_file_name,
spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_hoodie_partition_path, spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_hoodie_record_key,
spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_ac_header_id, spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_ac_line_id,
spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_ac_receive_cost_id, spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_accrual_status_code,
spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_ac_logical_is_deleted, spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_amount,
spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_accrual_time_code, spark_catalog.s999_sbi_stp_rtd_ac_transaction_cost_t_accrual_time_variance]
```

### 原因分析

使用Spark SQL写数据入hudi表，由于配置问题，表schema中会自动增加一列类型为array<string>，名称为col的列。

| 源字段        |    |           |  | 目的字段       |    |                          |                          |
|------------|----|-----------|--|------------|----|--------------------------|--------------------------|
| 名称         | 样值 | 类型        | 操作   | 名称         | 类型 | 分列                       | 操作                       |
| date1      |    | timestamp | <input checked="" type="checkbox"/> <input type="checkbox"/> | date1      |    | <input type="checkbox"/> | <input type="checkbox"/> |
| timestamp1 |    | timestamp | <input checked="" type="checkbox"/> <input type="checkbox"/> | timestamp1 |    | <input type="checkbox"/> | <input type="checkbox"/> |

### 解决方案

字段映射中删去此列，如果是自动建表，SQL中也要把此列删去。

## 3.58 Hudi 目的端的作业自动建表报错：schema 不匹配，建表失败怎么办？

### 问题描述

cdm迁移数据到hudi，hudi选择自动建表，执行建表语句报schema不匹配错误“org.apache.spark.sql.AnalysisException:Specified schema in create table statement is not equal to the table schema”。

### 原因分析

从metastore中删了表，但是文件没有清空，表的目录文件存在导致的，可能建的是外表。

### 解决方案

将表目录清空，再重新执行作业。

```

.../tomcat/webapps/sqoop/WEB-INF/classes/mrs/s000_cqrs_hinv_common_hudi_ai/user.keytab
2022-01-26 10:06:30.797 [ERROR] [cdm-job-submit-pool112] [o.apache.sqoop.common.SqoopException:118] [SqoopException
java.sql.SQLException: org.apache.hive.service.c11.HiveSQLException: ERROR running query: org.apache.spark.sql.AnalysisException: Specified schema in create table
statement is not equal to the table schema.You should not specify the schema for an exist table: `s000_cqrs_hinv_common`.`bas_inv_account_periods_t`
    at
    org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation.org$apache$spark$sql$hive$thriftserver$SparkExecuteStatementOperation$$execute(SparkExecuteSta
tementOperation.scala:387)
      at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2$$anon$3.$anonfun$run$3(SparkExecuteStatementOperation.scala:276)
      at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
      at org.apache.spark.sql.hive.thriftserver.SparkOperation.withLocalProperties(SparkOperation.scala:78)
      at org.apache.spark.sql.hive.thriftserver.SparkOperation.withLocalProperties$(SparkOperation.scala:62)
      at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation.withLocalProperties(SparkExecuteStatementOperation.scala:46)
      at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2$$anon$3.run(SparkExecuteStatementOperation.scala:276)
      at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2$$anon$3.run(SparkExecuteStatementOperation.scala:263)
      at java.security.AccessController.doPrivileged(Native Method)
      at javax.security.auth.Subject.doAs(Subject.java:422)
      at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1761)
      at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2.run(SparkExecuteStatementOperation.scala:290)
      at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
      at java.util.concurrent.FutureTask.run(FutureTask.java:266)
      at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:1149)
      at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:624)
      at java.lang.Thread.run(Thread.java:748)
Caused by: org.apache.spark.sql.AnalysisException: Specified schema in create table statement is not equal to the table schema.You should not specify the schema for
an exist table: `s000_cqrs_hinv_common`.`bas_inv_account_periods_t`
    at org.apache.spark.sql.hudi.analysis.HoodieResolveReferences$$anonfun$apply$1.applyOrElse(HoodieAnalysis.scala:320)
    at org.apache.spark.sql.hudi.analysis.HoodieResolveReferences$$anonfun$apply$1.applyOrElse(HoodieAnalysis.scala:123)

```

### 3.59 Hudi 目的端的作业失败，日志报错 Read Timeout 怎么解决？

#### 问题描述：

作业日志报错Read time out，日志如下：

```

... 20 common frames omitted
Caused by: java.sql.SQLException: Could not establish connection to jdbc:hive2://node-master9wQFL:22550/;pr
incipal=spark2x/hadoop.a4e2fe01_20e5_4298_95db_e0e775c19a0f.com@A4E2FE01_20E5_4298_95DB_E0E775C19A0F.COM;sa
slQop=auth-conf;saasl.qop=auth-conf;serviceDiscoveryMode=zooKeeper;auth=KERBEROS;socketTimeout=120;zooKeeper
Namespace=sparkthriftserver2x;user.principal=s000_cqrs_hah;user.keytab=/rds/cdm/tomcat/webapps/sqoop/WEB-INF
F/classes/mrs/s000_cqrs_hah_hudi/user.keytab: java.net.SocketTimeoutException: Read timed out
    at org.apache.hive.jdbc.HiveConnection.openSession(HiveConnection.java:931)
    at org.apache.hive.jdbc.HiveConnection.<init>(HiveConnection.java:259)
    ... 25 common frames omitted
Caused by: org.apache.thrift.transport.TTTransportException: java.net.SocketTimeoutException: Read timed out
    at org.apache.thrift.transport.TIOStreamTransport.read(TIOStreamTransport.java:127)
    at org.apache.thrift.transport.TTTransport.readAll(TTTransport.java:86)
    at org.apache.thrift.transport.TSaslTransport.readLength(TSaslTransport.java:365)
    at org.apache.thrift.transport.TSaslTransport.readFrame(TSaslTransport.java:450)
    at org.apache.thrift.transport.TSaslTransport.read(TSaslTransport.java:424)
    at org.apache.thrift.transport.TSaslClientTransport.read(TSaslClientTransport.java:38)
    at org.apache.thrift.transport.TTTransport.readAll(TTtransport.java:86)
    at org.apache.hadoop.hive.metastore.security.TFilterTransport.readAll(TFilterTransport.java:62)
    at org.apache.thrift.protocol.TBinaryProtocol.readAll(TBinaryProtocol.java:455)
    at org.apache.thrift.protocol.TBinaryProtocol.readI32(TBinaryProtocol.java:354)
    at org.apache.thrift.protocol.TBinaryProtocol.readMessageBegin(TBinaryProtocol.java:243)
    at org.apache.thrift.TServiceClient.receiveBase(TServiceClient.java:77)
    at org.apache.hive.service.rpc.thrift.TCLIService$Client.recv_OpenSession(TCLIService.java:149)
    at org.apache.hive.service.rpc.thrift.TCLIService$Client.OpenSession(TCLIService.java:136)
    at org.apache.hive.jdbc.HiveConnection.openSession(HiveConnection.java:912)
    ... 26 common frames omitted
Caused by: java.net.SocketTimeoutException: Read timed out
    at java.net.SocketInputStream.socketRead1(Native Method)
    at java.net.SocketInputStream.socketRead(SocketInputStream.java:116)
    at java.net.SocketInputStream.read(SocketInputStream.java:171)
    at java.net.SocketInputStream.read(SocketInputStream.java:141)
    at java.io.BufferedInputStream.fill(BufferedInputStream.java:246)
    at java.io.BufferedInputStream.read1(BufferedInputStream.java:286)
    at java.io.BufferedInputStream.read(BufferedInputStream.java:345)
    at org.apache.thrift.transport.TIOStreamTransport.read(TIOStreamTransport.java:125)
    ... 40 common frames omitted

```

## 原因分析

1. 确认MRS集群的JdbcServer是多实例模式还是多租模式。
  - 如果是多实例模式，跳转3。
  - 否则跳转2。
2. 多租户模式下，确认其他租户的作业是否正常。
  - 如果所有租户的作业执行spark sql都有问题，跳转3。
  - 否则，跳转4。
3. 进一步确认：用dlf建个脚本，选择直连接，执行一条spark sql，看是否报time out的错（甚至可能数据库都list不出来）。如果有以上现象，大概率是MRS集群的jdbc server出了问题。
4. 单租户执行不了spark sql，则多半是队列资源限制，打开yarn，搜索租户的队列，查看Spark2x-JDBCServer2x的yarn任务，此时可能会搜索不到yarn任务，或者State为ACCEPTED，这两种情况都是资源不足起不了yarn任务的现象。打开yarn的schedule，查看队列资源，关注以下几个参数：

*Used Resources: 已使用的内存与CPU核数*

*Max Resources: 队列中最大可供使用的内存与CPU核数*

*Used Application Master Resources: 已使用的AM资源*

*Max Application Master Resources: 队列中最大可供使用的AM资源*

通过对比基本就能确定是哪个资源不足导致yarn任务执行异常。

| ID                             | User    | Name                             | Application Type | Queue           | Application Priority | StartTime                     | FinishTime | State   | FinalStatus | Running Containers | Allocated CPU V-Cores | Allocated Memory MB | Allocated GPUs | Reserved CPU V-Cores | Reserved Memory MB | Reserved GPUs | % of Queue | % of Cluster |
|--------------------------------|---------|----------------------------------|------------------|-----------------|----------------------|-------------------------------|------------|---------|-------------|--------------------|-----------------------|---------------------|----------------|----------------------|--------------------|---------------|------------|--------------|
| application_1648970769137_6463 | spark2x | Spark2x-JDBCServer2x-7213.32.174 | SPARK            | s000_sbi_invest | 0                    | Sat Apr 9 10:36:18 +0800 2022 | N/A        | RUNNING | UNDEFINED   | 1                  | 1                     | 20480               | 0              | 0                    | 0                  | 0             | 6.5        | 0.0          |

| 'root.s000_sbi_ar' Queue Status          |                              |
|--|------------------------------|
| Num Active Applications:                 | 15                           |
| Num Pending Applications:                | 0                            |
| Resource Pool:                           | [default]                    |
| Default Node Label Expression:           |                              |
| Open Status:                             | OPEN                         |
| Active Status:                           | ACTIVE                       |
| Used Resources:                          | <memory:376832, vCores:117>  |
| Min Resources:                           | <memory:1585152, vCores:387> |
| Max Resources:                           | <memory:1585152, vCores:387> |
| Reserve Resources:                       | <memory:0, vCores:0>         |
| Configured Max Application Master Limit: | 50.0                         |
| Max Application Master Resources:        | <memory:792576, vCores:193>  |
| Used Application Master Resources:       | <memory:71680, vCores:15>    |
| Num Allocated Containers:                | 44                           |
| Num Pending Containers:                  | 0                            |
| Instantaneous Resources:                 | <memory:376832, vCores:117>  |
| Alloc Order Policy:                      | FIFO                         |
| Steady Share:                            | 2.5%                         |
| Max Resource Share:                      | 2.5%                         |
| Instantaneous Share:                     | 0.6%                         |
| Disaster Recovery Migration:             | false                        |

## 解决方案

扩充队列资源，或者停止其他yarn任务释放资源。

## 3.60 Hudi 目的端的作业执行卡 Running，读取行数写入行数相等且不再增加怎么解决？

### 问题描述

CDM写Hudi为两段式，先写到hive临时表，然后再执行spark sql写到Hudi，写入行数统计的是写hive临时表的行数，当行数不再增长时，说明源端数据已经读完写到Hive表中，此时作业正在执行Spark SQL过程中，需要等Spark SQL执行完作业才会结束。

## 原因分析

打开日志，搜索insert into，找到如下的日志，根据日志中打印的Yarn ApplicationId到MRS Resource Manager上看Yarn任务详情。

```
2022-09-14 01:10:13.574|INFO |Write-Hoodie|[org.apache.zookeeper.ZooKeeper:878]|Initiating client connection, connectionString=node-master1atyh:2181,node-
master2bph:2181,node-master3frhr:2181 sessionTimeout=60000 watcher=org.apache.curator.ConnectionState@55212115
2022-09-14 01:10:13.575|INFO |Write-Hoodie|[o.apache.zookeeper.ClientCnxnSocket:238]|jute.maxBuffer value is 4194204 Bytes
2022-09-14 01:10:13.575|INFO |Write-Hoodie|[org.apache.zookeeper.ClientCnxn:1706]|zookeeper.request.timeout value is 120000. feature.enabled=true
2022-09-14 01:10:13.575|INFO |Write-Hoodie|[o.a.z.c.w.ClientBindingHelper:94]|zookeeper.client.bind.port.range is not configured.
2022-09-14 01:10:13.575|INFO |Write-Hoodie|[o.a.z.c.w.ClientBindingHelper:61]|zookeeper.client.bind.address is not configured.
2022-09-14 01:10:13.636|INFO |Write-Hoodie|[o.a.c.f.imps.CuratorFrameworkImpl:386]|Default schema
2022-09-14 01:10:13.882|INFO |Write-Hoodie|[org.apache.zookeeper.ZooKeeper:1457]|connectionid: 0x160026a296841510 closed
2022-09-14 01:10:13.883|INFO |Write-Hoodie|[org.apache.hive.jdbc.HiveConnection:1814]|Login timeout is 300000
2022-09-14 01:10:13.907|INFO |Write-Hoodie|[org.apache.hive.jdbc.HiveConnection:377]|user login success.
2022-09-14 01:10:13.913|INFO |Write-Hoodie|[org.apache.hive.jdbc.HiveConnection:476]|will try to open client transport with JDBC Uri: jdbc:hive2://node-
master3FRHR:22550/?principal=spark2x/hadoop.9f15c4d7_8a23_45ec_b8bc_370df18bd9f6.com@9f15c4d7_8a23_45ec_b8bc_370df18bd9f6.COM;sslqop=auth-conf;ssl.qop=auth-
conf;serviceDiscoveryMode=zooKeeper;auth=KERBEROS;socketTimeout=300;zooKeeperNamespace=sparkthriftserver2x;user.principal=jack;user.keytab=/rds/cdm/tomcat/webapps/sqo
op/WEB-INF/classes/mrs/hudi_link/user.keytab
2022-09-14 01:10:14.086|INFO |Write-Hoodie|[o.a.s.c.spark.util.SparkJdbcUtils:90]|Connect to: Spark SQL(3.1.1-hw-ei-312032), Yarn Application Id =
application_1663034201663_0254 | Driver: Hive JDBC(3.1.0-hw-ei-312005)
2022-09-14 01:10:14.086|INFO |Write-Hoodie|[o.a.s.c.spark.model.SparkStatement:66]|execute sql: insert into `jack`.`performance_big_1g` select `par_no`,`p1`,`p2` from
`jack`.`performance_big_1g_tmp_51f5e4c85f2447122f49b2ee088afaf1`
```

### 说明

执行Spark SQL的速度与租户队列资源强相关，在执行Hudi任务前，请确保租户队列资源充足。

## 3.61 Hudi 目的端的作业执行卡 Running，数据写入行数一直显示为 0 如何处理？

### 问题描述

打开日志，最后一行日志如下所示，则说明此时集群并发资源消耗殆尽，或者集群内存使用达到阈值，新提交的作业需要排队等待。

```
submit task attempt_local1847334969_1748_m_000003_0, current waiting task number for job
job_local1847334969_1748 is : 4
```

### 原因分析

- 集群并发数到达上限  
联系SRE查看cdm后台日志：/var/log/cdm/local/framework.log，搜索关键字：cluster running task，如果运行的并发数与available的并发数一致，则说明此时并发数已到达集群上限。
- 集群内存使用达到阈值  
联系SRE查看cdm后台日志：/var/log/cdm/local/framework.log，搜索关键字：memory usage exceeds threshold，如果此时集群在不断打印此日志，则说明堆内存使用已经超过75%，集群可能有oom的风险。

### 解决方案

调整作业并发数，使其不超过集群并发数（建议集群并发数不超过46）。集群并发数即配置管理页面的最大抽取并发数。



## 3.62 Hudi 目的端的作业执行 Spark SQL 写入 Hudi 失败怎么办？

### 问题描述

- 报错：hoodie table path not found。

```
2022-01-25 22:47:26.494[ERROR]Thread-14[io.apache.sqoop.common.SqoopException:118]SqoopException
java.sql.SQLException: org.apache.hive.service.cli.HiveSQLException: ERROR running query: org.spark_project.guava.util.concurrent.UncheckedExecutionException:
org.apache.hudi.exception.TableNotFoundException: Hoodie table not found in path Unable to find a hudi table for the user provided paths.
    at
    org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation.org$apache$spark$sql$hive$thriftserver$SparkExecuteStatementOperation$$execute(SparkExecuteStat
ementOperation.scala:387)
    at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2$$anon$3.$anonfun$run$3(SparkExecuteStatementOperation.scala:276)
    at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
    at org.apache.spark.sql.hive.thriftserver.SparkOperation.withLocalProperties(SparkOperation.scala:78)
    at org.apache.spark.sql.hive.thriftserver.SparkOperation.withLocalProperties$(SparkOperation.scala:62)
    at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation.withLocalProperties(SparkExecuteStatementOperation.scala:46)
    at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2$$anon$3.run(SparkExecuteStatementOperation.scala:276)
    at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2$$anon$3.run(SparkExecuteStatementOperation.scala:263)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1761)
    at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2.run(SparkExecuteStatementOperation.scala:290)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
```

- 报错：写入记录中存在空值，写入失败。

```
Caused by: org.apache.spark.SparkException: Job aborted due to stage failure:
Aborting TaskSet 15.0 because task 0 (partition 0)
cannot run anywhere due to node and executor excludeOnFailure.
Most recent failure:
Lost task 0.0 in stage 15.0 (TID 14) (node-group-1xn1K0001 executor 5): org.apache.hudi.exception.HoodieKeyException: recordKey values: "numeric_dec2:___null___" for
fields: [numeric_dec2] cannot be entirely NULL or empty.
    at org.apache.hudi.keygen.KeyGenUtils.getRecordKey(KeyGenUtils.java:109)
    at org.apache.hudi.keygen.ComplexAvroKeyGenerator.getRecordKey(ComplexAvroKeyGenerator.java:43)
    at org.apache.hudi.keygen.ComplexKeyGenerator.getRecordKey(ComplexKeyGenerator.java:49)
    at org.apache.spark.sql.hudi.command.SqlKeyGenerator.getRecordKey(SqlKeyGenerator.scala:62)
    at org.apache.hudi.keygen.BaseKeyGenerator.getKey(BaseKeyGenerator.java:62)
    at org.apache.hudi.HoodieSparkSqlWriter$.anonfun$write$7(HoodieSparkSqlWriter.scala:237)
    at scala.collection.Iterator$$anon$10.next(Iterator.scala:459)
    at org.apache.spark.storage.memory.MemoryStore.putIterator(MemoryStore.scala:222)
    at org.apache.spark.storage.memory.MemoryStore.putIteratorAsBytes(MemoryStore.scala:349)
    at org.apache.spark.storage.BlockManager.$anonfun$doPutIterator$1(BlockManager.scala:1442)
    at org.apache.spark.storage.BlockManager.org$apache$spark$storage$BlockManager$$doPut(BlockManager.scala:1352)
    at org.apache.spark.storage.BlockManager.doPutIterator(BlockManager.scala:1416)
    at org.apache.spark.storage.BlockManager.getOrEraseUpdate(BlockManager.scala:1239)
    at org.apache.spark.rdd.RDD.getOrCompute(RDD.scala:384)
    at org.apache.spark.rdd.RDD.iterator(RDD.scala:335)
    at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:52)
    at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:373)
    at org.apache.spark.rdd.RDD.iterator(RDD.scala:337)
    at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:52)
    at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:373)
    at org.apache.spark.rdd.RDD.iterator(RDD.scala:337)
```

- 报错：killed by external signal。

```

at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation.org$apache$spark$sql$hive$thriftserver$SparkExecuteStatementOperation$$execute(SparkExecuteStatementOperation.scala:347)
... 16 more
Caused by: org.apache.spark.SparkException: Job aborted due to stage failure: Task 164 in stage 17_0 failed 4 times, most recent failure: Lost task 164.3 in stage 17_0 (TID 3734) (node-group-1y6mP0018 executor 278): ExecutorLostFailure (executor 278 exited caused by one of the running tasks) Reason: Container from a bad node: container_e04_1639218193738_0973_01_000324 on host: node-group-1y6mP0018. Exit status: 143. Diagnostics: [2021-12-22 19:39:19.580]Container killed on request. Exit code is 143
[2021-12-22 19:39:19.580]Container exited with a non-zero exit code 143.
[2021-12-22 19:39:19.581]Killed by external signal
.
Driver stacktrace:
at org.apache.spark.scheduler.DAGScheduler.failJobAndIndependentStages(DAGScheduler.scala:2298)
at org.apache.spark.scheduler.DAGScheduler.$anonfun$abortStage$2(DAGScheduler.scala:2247)
at org.apache.spark.scheduler.DAGScheduler.$anonfun$abortStage$2$adapted(DAGScheduler.scala:2246)
at scala.collection.mutable.ResizableArray.foreach(ResizableArray.scala:62)
at scala.collection.mutable.ResizableArray.foreach$(ResizableArray.scala:55)
at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:49)
at org.apache.spark.scheduler.DAGScheduler.abortStage(DAGScheduler.scala:2246)
at org.apache.spark.scheduler.DAGScheduler.$anonfun$handleTaskSetFailed$1(DAGScheduler.scala:1119)
at org.apache.spark.scheduler.DAGScheduler.$anonfun$handleTaskSetFailed$1$adapted(DAGScheduler.scala:1119)
at scala.Option.foreach(Option.scala:407)
at org.apache.spark.scheduler.DAGScheduler.handleTaskSetFailed(DAGScheduler.scala:1119)
at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.doOnReceive(DAGScheduler.scala:2485)

```

- 报错：java.lang.IllegalArgumentException。

```

2022-05-27 15:36:46.179] ERROR [Thread-24088][o.apache.sqoop.common.SqoopException:118][SqoopException
java.sql.SQLException: org.apache.hive.service.cli.HiveSQLException: ERROR running query: java.lang.IllegalArgumentException
at
org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation.org$apache$spark$sql$hive$thriftserver$SparkExecuteStatementOperation$$execute(SparkExecuteStatementOperation.scala:387)
at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2$anon$3.$anonfun$run$3(SparkExecuteStatementOperation.scala:276)
at scala.runtime.java8.JFunction$mcV$sp.apply(JFunction$mcV$sp.java:23)
at org.apache.spark.sql.hive.thriftserver.SparkOperation.withLocalProperties(SparkOperation.scala:78)
at org.apache.spark.sql.hive.thriftserver.SparkOperation.withLocalProperties$(SparkOperation.scala:62)
at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation.withLocalProperties(SparkExecuteStatementOperation.scala:46)
at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2$anon$3.run(SparkExecuteStatementOperation.scala:276)
at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2$anon$3.run(SparkExecuteStatementOperation.scala:263)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:422)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1761)
at org.apache.spark.sql.hive.thriftserver.SparkExecuteStatementOperation$$anon$2.run(SparkExecuteStatementOperation.scala:290)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)
Caused by: java.lang.IllegalArgumentException
at org.apache.hudi.common.util.ValidationUtils.checkArgument(ValidationUtils.java:37)
at org.apache.hudi.common.table.timeline.HoodieActiveTimeline.transitionState(HoodieActiveTimeline.java:429)
at org.apache.hudi.common.table.timeline.HoodieActiveTimeline.transitionState(HoodieActiveTimeline.java:410)
at org.apache.hudi.common.table.timeline.HoodieActiveTimeline.saveAsComplete(HoodieActiveTimeline.java:168)
at org.apache.hudi.client.AbstractHoodieWriteClient.commit(AbstractHoodieWriteClient.java:269)
at org.apache.hudi.client.AbstractHoodieWriteClient.commitStats(AbstractHoodieWriteClient.java:214)
at org.apache.hudi.client.SparkRDDWriteClient.commitForWrite(SparkRDDWriteClient.java:142)

```

## 原因分析

- 表在metastore中存在，但不是Huid表，或者表在metastore中存在，但是表目录不存在，根因是在表目录下没有.hoodie目录。可能删表的时候只删了文件而没有drop table。
- 设置为主键或者预聚合键的列有空值，写入hoodie会失败。  
查看作业配置，查看表属性中hoodie.datasource.write.recordkey.field、hoodie.datasource.write.precombine.field、hoodie.datasource.write.partitionpath.field配置的列在源端数据中是否存在空值。
- 可能数据倾斜导致executor使用内存超出限制，具体原因需要联系MRS定位。Yarn Application ID可以从日志中获取，日志搜索“Yarn Application Id”关键字，查询离报错信息最近的Yarn Application ID即可。

### 自主排查方式

- a. 登录yarn，根据applicationId查询到yarn任务，打开ApplicationManager。



- b. 打开stage->查看fail状态的task，通过日志或者界面显示，可以看到失败原因，通常会以下报错：

transferring unroll memory to storage memory failed ( RDD中缓存超过了executor的内存out of memory ) 。

Tasks (394)

Show 20 entries

| Index | Task ID | Attempt | Status | Locality level | Executor ID | Host                 | Logs   | Launch Time         | Duration | GC Time | Input Size / Records | Shuffle Write Size / Records | Spill (Memory) | Spill (Disk) | Errors  |
|-------|---------|---------|--------|----------------|-------------|----------------------|--------|---------------------|----------|---------|----------------------|------------------------------|----------------|--------------|---|
| 5     | 5       | 0       | FAILED | PROCESS_LOCAL  | 4           | node-group-12k0045   | stdout | 2022-04-16 14:38:03 | 3.1 min  | 4 s     | 100.3 MB / 12933848  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 12    | 12      | 0       | FAILED | PROCESS_LOCAL  | 6           | node-group-1jxk0010  | stdout | 2022-04-16 14:38:04 | 3.1 min  | 5 s     | 100.8 MB / 12917926  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 19    | 19      | 0       | FAILED | PROCESS_LOCAL  | 15          | node-group-12z0009   | stdout | 2022-04-16 14:38:05 | 3.2 min  | 5 s     | 108.7 MB / 13945254  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 26    | 26      | 0       | FAILED | PROCESS_LOCAL  | 13          | node-group-13pK0008  | stdout | 2022-04-16 14:38:05 | 3.1 min  | 6 s     | 100.6 MB / 12900046  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 238   | 238     | 0       | FAILED | PROCESS_LOCAL  | 10          | node-group-12z00048  | stdout | 2022-04-16 14:38:11 | 2.7 min  | 4 s     | 93.6 MB / 11602902   |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 244   | 244     | 0       | FAILED | PROCESS_LOCAL  | 20          | node-group-1ac00033  | stdout | 2022-04-16 14:38:11 | 3.0 min  | 5 s     | 102.7 MB / 13176840  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 251   | 251     | 0       | FAILED | PROCESS_LOCAL  | 27          | node-group-12z00024  | stdout | 2022-04-16 14:38:11 | 3.2 min  | 12 s    | 107.3 MB / 13768029  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 263   | 263     | 0       | FAILED | PROCESS_LOCAL  | 25          | node-group-13pK0002  | stdout | 2022-04-16 14:38:11 | 3.0 min  | 4 s     | 102.7 MB / 13158791  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 293   | 293     | 0       | FAILED | PROCESS_LOCAL  | 61          | node-group-12z00002  | stdout | 2022-04-16 14:38:12 | 3.5 min  | 24 s    | 105 MB / 13473050    |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 300   | 300     | 0       | FAILED | PROCESS_LOCAL  | 38          | node-group-13pK00024 | stdout | 2022-04-16 14:38:12 | 3.6 min  | 39 s    | 103.2 MB / 13241231  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 303   | 303     | 0       | FAILED | PROCESS_LOCAL  | 34          | node-group-12k00050  | stdout | 2022-04-16 14:38:12 | 3.6 min  | 11 s    | 124.5 MB / 15163187  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 306   | 306     | 0       | FAILED | PROCESS_LOCAL  | 51          | node-group-13pK00017 | stdout | 2022-04-16 14:38:12 | 4.1 min  | 36 s    | 126.3 MB / 15249400  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 313   | 313     | 0       | FAILED | PROCESS_LOCAL  | 16          | node-group-12z00010  | stdout | 2022-04-16 14:38:12 | 3.2 min  | 5 s     | 106.2 MB / 13635801  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 320   | 320     | 0       | FAILED | PROCESS_LOCAL  | 47          | node-group-12z00011  | stdout | 2022-04-16 14:38:12 | 2.9 min  | 4 s     | 101.7 MB / 13000029  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |
| 327   | 327     | 0       | FAILED | PROCESS_LOCAL  | 42          | node-group-13pK00029 | stdout | 2022-04-16 14:38:12 | 3.0 min  | 4 s     | 104.1 MB / 13354671  |                              |                |              | java.lang.AssertionError: assertion failed: transferring unroll memory to storage memory failed |

- Hudi不支持并发写，会产生commit冲突。

## 解决方案

- 在DataArts Studio或者Hue或者spark-beeline上执行drop table将表从metastore中删除，然后作业配置“不存在时创建”重跑作业。或者删除后自己执行建表语句重建一个Hudi表。

### 📖 说明

对于MOR表来说，删表需要把ro与rt表也同时删除。否则会出现schema残留的问题。

- 删除空值后重跑作业。
- 具体办法：
  - 在作业管理界面选择“更多-失败重试”，尝试重新执行Spark SQL。
  - 通过DataArts Studio执行Spark SQL，设置执行参数或者调整SQL。

调整Spark切片大小：

set spark.sql.files.maxPartitionBytes=xxM；默认值为128M，可适当调整为64M或者32M。

如果数据切分不均匀，可以修改SQL配置DISTRIBUTE BY rand()，增加一个shuffle过程，打散数据（需要占用较多资源，资源不多时慎用）。

insert into xx select \* from xxx DISTRIBUTE BY rand();

- 使用DataArts Studio API方式提交Spark SQL，调大executor内存。
- 排查是否有其他连接在同时写hudi表，如果有，将连接停止，然后CDM作业失败重试。

## 3.63 Hudi 目的端的作业执行过程中，由于源端连接闪断、超时或主动终止连接导致作业执行失败怎么处理？

### 问题描述

作业执行过程中，由于源端连接闪断、超时或者源端主动终止了连接导致作业执行失败。

### 问题分析

日志中出现源端的read timeout报错，或者terminate by xxx之类的报错。

### 解决方案

- 如果源端网络不稳定，可以使用分片重试能力多次执行作业，可能需要调整作业配置。
- 如作业配置了分片数，或者源端为分区表，且作业配置了按表分区抽取，则单击更多-分片重试，重跑失败分片（比如配置了100个分片，上次执行到50个分片报错，则单击失败重试后，仅会执行剩余50个分片）。
- 如且源端非分区表，作业未配置分片数，建议调大作业分片数，再重新执行作业，后续再发生异常通过失败重试断点续传。
- 如源端为分区表，且未配置按表分区抽取，建议配置按表分区抽取后，重新执行作业，后续再发生异常通过失败重试断点续传。