

大数据基础服务与创新示范应用建设项目

ETL 全量+增量上云实施操作手册

V0.10

文档敏感性定义	敏感		
编写人	神夕	编写日期	2018-03-31
审核人	个人	审核日期	2018-03-31
公开范围			
建设单位			
承建单位			

修订状况

章节 编号	章节名称	修订内容简述	修订人	修订日期	修订前 版本号	修订后 版本号	批准人

目录

1. 简介	4
1.1. 目的	4
1.2. 范围	4
1.3. 定义、业务术语、缩略语	4
1.4. 参考资料	4
2. DATAX+OGG 全量上云实施	4
2.1. OGG 源端目标端配置流程	4
2.1.1. OGG 简介	4
2.1.1.1. 相关术语	4
2.1.1.2. 原理特点	6
2.1.1.3. 安装使用	6
2.1.1.3.1. 下载安装	6
2.1.1.3.2. 双端配置	8
2.1.1.3.3. 测试 OGG	14
2.2. DATAX 全量初始化实施	16
2.2.1. 总体流程及结构	16
2.2.1.1. 流程图	16
2.2.1.2. 文件目录结构	17
2.2.2. 初始化配置	19
2.2.2.1. 配置 ty_datasource.conf	19
2.2.2.2. 配置 ty_createJson_ql.conf	20
2.2.2.3. odps 建表	21
2.2.2.3.1. 源表与目标表字段类型映射	21
2.2.2.3.2. 生成建表脚本	21
2.2.2.3.3. odps 建表	23
2.2.2.4. 生成 Json 文件	24
2.2.3. 提交任务上云	25
2.2.3.1. 统计源表记录数	25
2.2.3.2. 提交 datax 任务	25
2.2.4. 结果统计(非必须)	28
2.2.5. 合并分区	30
2.3. DATAX 增量初始化实施	34
2.3.1. 总体流程及结构	34
2.3.1.1. 流程图	34
2.3.1.2. 具体步骤	35
2.3.1.3. 补数据	41
2.3.1.4. 分析失败任务	43

1. 简介

1.1. 目的

1.2. 范围

1. 适用范围：ORACLE 到 ODPS 全量数据上云项目；
2. 阅读对象：
 - (1) 本项目局方项目组；
 - (2) 本项目乙方项目组；

1.3. 定义、业务术语、缩略语

1.4. 参考资料

- 《OGG MaxCompute 插件.pdf》；
- 《OGG 环境搭建手册.pdf》；
- 《Oracle Golden Gate 图文并茂快速掌握.pdf》；
- 《OGG 部署配置及运维方案.docx》

2. DataX+ogg 全量+增量上云实施

2.1. Ogg 源端目标端配置流程

2.1.1. OGG 简介

Oracle Golden Gate 是 Oracle 旗下一款支持异构平台之间高级复制技术，是 Oracle 力推一种 HA 高可用产品，简称“OGG”，可以实现 Active-Active 双业务中心架构

2.1.1.1. 相关术语

OracleGolden Gate 有源端和目标端，源端捕获日志发送到目标端应用，这个过程分为六步骤

- 捕获：实时捕获交易日志(已提交数据)，包含 DML 和 DDL，并可根据规则进行过滤
- 队列：把捕获的日志数据加载入队列(写入 trail 文件)，这是可选项，为了提高安全性，怕网络传丢了。也可以不入队列，直接从 redo buffer 传递给目标端
- 数据泵：将 trail 文件广播到不同的目标端
- 网络：从源网络压缩加密后传送到目的网络
- 接收队列：接收从源端传过来的 trail 文件
- 交付：把 trail 文件内容转换成 SQL 语句在目标库执行 双向复制：在把另一端重新配置成源端，即可实现双向复制，这就是 Active-Active 双业务中心

Golden Gate 进程

- Manager 进程：这是 GG 全局主进程，它是 GG 守护进程统筹全局，它可以启动、监控、终止 Golden Gate 的其它进程，收集错误报告及事件，分配数据存储空间，发布阈值告警等，在源端和目标端有且只有一个 Manager 进程。
- Extract 进程：运行在源端的进程，实时捕获交易数据，可以直接在 redo buffer 捕获传递到目标端，也可以在 redo buffer 捕获先写入 trail 队列在传递到目标端。非 Oracle 库支持从数据表捕获数据。
- Pump 进程：运行在源端的进程，将源端产生的本地 trail 文件广播到不同的目标端，pump 进程本质是 extract 进程的一种特殊形式，如果不使用 trail 文件，那么 extract 进程在捕获完交易日志后直接传递到目标端，生成远程 trail 文件。
- Collector 进程：运行在目标端的进程，专门接收从源端传过来的 trail 文件日志生成队列。
- Delivery 进程：运行在目标端的进程，通常我们也把它叫做 replicat 进程，是数据传递的最后一站，负责读取远程 trail 文件内容，解析为 SQL 语句在目标库上执行。

2.1.1.2. 原理特点

- 实时数据复制
- 异构平台数据同步
- 支持断点续传，不影响系统连续运行
- 高性能，属于轻量级软件
- 保证数据引用完整性和事物一致性
- 整合 ETL Tools Message Service
- 灵活拓扑结构 1:1 1:N N:1 N:N 双向复制
- 复制冲突检测 and 解决
- 支持数据压缩和加密
- TCP/IP WAN LAN
- 根据事务大小和数量自动管理内存
- 支持多活业务中心
- 以交易数据为单位复制，保证交易一致性
- 支持数据过滤和转换，可自定义基于表和行的过滤规则，实时在异构环境下转换数据

2.1.1.3. 安装使用

2.1.1.3.1. 下载安装

安装 OGG

需要下载两个文件，源端安装 OGG，目标端安装 Adapter

安装包下载

<http://www.oracle.com/technetwork/middleware/goldengate/downloads/index.html>

然后解压安装，选择静默安装

2.1.1.3.1.1. 源端安装

源端配置/response/oggcore.rsp 文件

目前oracle一般采取response安装的方式，在response/oggcore.rsp中配置安装依赖，具体如下：

```
1. oracle.install.responseFileVersion=/oracle/install/rspfmt_ogginstall_response_schema_v12_1_2
2. # 需要目前与oracle版本对应
3. INSTALL_OPTION=ORA11g
4. # goldegate主目录
5. SOFTWARE_LOCATION=/home/oracle/u01/ggate
6. # 初始不启动manager
7. START_MANAGER=false
8. # manger端口
9. MANAGER_PORT=7839
10. # 对应oracle的主目录
11. DATABASE_LOCATION=/home/oracle/u01/app/oracle/product/11.2.0/dbhome_1
12. # 暂可不配置
13. INVENTORY_LOCATION=
14. # 分组（目前暂时将oracle和ogg用同一个账号ogg_test，实际可以给ogg单独账号）
15. UNIX_GROUP_NAME=oinstall
```

有两个需要注意的地方

INSTALL_OPTION=ORA11g --oracle 版本是多少就填多少

UNIX_GROUP_NAME=oinstall --这个组名需要填对，可以通过 ll 命令查看文件夹属性组名并与之对应，否则会报错，还有每次重新安装时都要把以前生成的文件全部 del 掉，不然会报错

静默安装步骤：

./runInstaller -silent -responseFile
{YOUR_OGG_INSTALL_FILE_PATH}/response/oggcore.rsp

具体操作参见

https://help.aliyun.com/document_detail/28294.html?spm=5176.doc28291.6.593.mdVnrX

2.1.1.3.1.2. 目标端安装

目标端安装比较方便，解压 Adapter 出来即可。记得把整个文件夹目录以及包含文件的用户和组调成 oracle 用户下的，方便后续权限管理使用

2.1.1.3.2. 双端配置

2.1.1.3.2.1. 源端数据库配置

以 dba 身份进入数据库: sqlplus / as sysdba

创建独立的表空间, 这个路径根据自己安装的 oracle 路径填写

```
create tablespace ATMV datafile '/home/oracle/app/ATMV.dbf' size 1000m
autoextend on next 50m maxsize unlimited;
```

创建 ogg_test(可自行设定)用户, 密码为 ogg_test(可自行设定), 也可以不用创建新用户, 但是一定要给使用 OGG 的用户赋予 OGG 需要使用的权限

```
create user ogg_test identified by ogg_test default tablespace ATMV;
```

给 ogg 用户赋予权限以便 ogg 正常使用(有些权限在 9i 里没有)

```
GRANT CREATE SESSION ,ALTER SESSION , RESOURCE,CONNECT,SELECT ANY
DICTIONARY,SELECT ANY TRANSACTION TO OGG_TEST;
```

```
GRANT FLASHBACK ANY TABLE,SELECT ANY TABLE TO OGG_TEST;
```

```
GRANT EXECUTE ON dbms_flashback to OGG_TEST;
```

```
GRANT GGS_GGSUSER_ROLE to OGG_TEST;
```

检查附加日志情况, 若是返回 YES 则代表日志已开启最小补全日志, 若不是 YES, 则执行下面的步骤, 执行完之后要重新确认都返回 YES(部分日志情况 9i 没有)

```
Select    SUPPLEMENTAL_LOG_DATA_MIN,    SUPPLEMENTAL_LOG_DATA_PK,
SUPPLEMENTAL_LOG_DATA_UI,                SUPPLEMENTAL_LOG_DATA_FK,
SUPPLEMENTAL_LOG_DATA_ALL from v$database;
```

增加数据库附加日志, 可以指示数据库在日志中添加额外信息到日志流中, 以支持基于日志的工具如 ogg, 帮助 ogg 目标端分析识别修改的数据

```
alter database add supplemental log data;
```

```
alter database add supplemental log data (primary key, unique,foreign key)
columns;
```

全字段模式, 开启补全日志

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

开启数据库强制日志模式, 无论什么操作都进行 redo 的写入, 一些

nologging 的操作也会写如日志

```
alter database force logging;
```

```
#添加主键附加日志
```

```
alter table sys.seq$ add supplemental log data (primary key) columns;
```

执行 marker_setup.sql 脚本，会让你填写用户名，这些脚本在 ogg 源端安装文件夹里

```
@marker_setup.sql
```

```
@ddl_setup.sql
```

```
@role_setup.sql
```

```
# 执行脚本，开启 DDL trigger
```

```
@ddl_enable.sql
```

```
# 执行优化脚本
```

```
@ddl_pin ogg_test
```

```
# 安装 sequence support 序列支持
```

```
@sequence.sql
```

2.1.1.3.2.2.源端 mgr 配置

安装好之后通过./ggsci 进入命令行模式

创建必须目录 GGSCI> create subdirs

源端配置要配置好 mgr(进程管理)，extract(抽取进程)和 pump(投递进程)

➤ 配置 mgr

```
GGSCI> edit params mgr
```

```
PORT 7839 --端口号可以随意配置，本机传本机端口号要不同
```

```
DYNAMICPORTLIST 7840-8000 --允许的端口数量
```

```
AUTORESTART ER *, RETRIES 5, WAITMINUTES 3 --每隔 x 分钟自动开启进程
```

```
PURGEOLDEXTRACTS ./dirdat/*, USECHECKPOINTS, MINKEEPDAYS 7
```

```
--trail 文件推送位置，设置使用过后的 trail 文件的保留时间
```

```
LAGREPORTHOURS 1
```

```
LAGINFOMINUTES 30
```

```
LAGCRITICALMINUTES 45 --必选的三个参数，出现 report, info 等的时间
```

- 启动 mgr: start mgr
进程的运行日志在 dirrpt 文件夹下
- 查看 mgr 状态: info mgr
- 查看 mgr 配置: view params mgr

2.1.1.3.2.3. 源端 extract 配置

- 配置并增加进程 extract (用核心征管库举例 HX)

GGSCI> add ext hxn1,tranlog, begin now --如果是 9i 则需要指定 thread 1, thread 2

GGSCI> add extract ./dirdat/no , extract hxn1

extract hxn1

SETENV (ORACLE_SID="xxx")

SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)

Userid xxx,PASSWORD xxx --这三个与源端数据库相同

REPORT AT 01:59

REPORTROLLOVER AT 02:00 --设定切换一个日志的时间和间隔

CACHEMGR, CACHESIZE 256MB --用于控制存放未提交事务的虚拟内存

EXTTRAIL ./dirdat/no

NUMFILES 3000

EOFDELAYCSECS 30 --读到日志文件末尾时的休眠时间，缩小增强实时性

GETTRUNCATES --复制 TRUNCATE 操作

TRANLOGOPTIONS DBLOGREADER --指定在解析数据库日志时所需要的特殊参数，本例指定登陆人

DYNAMICRESOLUTION --动态处理解决

BR BRINTERVAL 2H , BRDIR BR --进程从恢复到其停止的时间点并恢复正常处理所需要的时间设定了一个时间上限

GETUPDATEBEFORES --得到修改之前的值

NOCOMPRESSDELETES --参数可以记录所有列删除值

TABLEEXCLUDE *.MLOG*;

TABLE ogg_test.*;

2.1.1.3.2.4.源端 pump 配置

- 配置并增加投递进程 phxnI

```
GGSCI> add ext phxnI,exttrailsource ./dirdat/no
```

```
GGSCI> add rmttrail ./dirdat/no , extract phxnI
```

```
extract phxnI
```

```
SETENV (ORACLE_SID="xxx")
```

```
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
```

```
Userid xxx,PASSWORD xxx
```

```
REPORT AT 01:59
```

```
REPORTROLLOVER AT 02:00
```

```
CACHEMGR, CACHESIZE 256MB
```

```
FLUSHCSECS 30 --冲刷时间设定
```

```
NUMFILES 3000
```

```
EOFDELAYCSECS 30
```

```
RMTHOST xx.xx.xx.xxx,MGRPORT xxxx
```

```
RMTTRAIL ./dirdat/no
```

```
GETTRUNCATES
```

PASSTHRU --必选参数,让 OGG 以直通模式运行,不必再从数据库查找表定义,故此参数要求双端的表名称,表结构必须一致

```
DYNAMICRESOLUTION
```

```
GETUPDATEBEFORES
```

```
NOCOMPRESSDELETES
```

```
TABLEEXCLUDE *.MLOG*;
```

```
TABLE ogg_test.*;
```

2.1.1.3.2.5.生成 def 表定义文件

- 编辑 defgen

```
GGSCI> edit params defgen
```

```
DEFSFILE ./dirdef/ogg.def
```

```
USERID xxx, PASSWORD xxx
```

```
table ogg_test.*;
```

- 在 shell 中执行如下命令，生成 ogg.def，并把这个 ogg.def 拷贝到目标端 dirdef 下

```
./defgen paramfile ./dirprm/defgen.prm
```

2.1.1.3.2.6. 目标端配置并开启

- 解压 Ada 文件并运行 ./ggsci

- GGSCI>create subdirs

- 配置 mgr

```
GGSCI> edit param mgr
```

```
PORT 7809 -- 端口号可以随意配置
```

```
dynamicportlist 8100-8200
```

```
autorestart er *, retries 5, waitminutes 3
```

```
purgeoldextracts ./dirdat/*,usecheckpoints, minkeepdays 10
```

```
LAGREPORTHOURS 5
```

```
LAGINFOMINUTES 10
```

```
LAGCRITICALMINUTES 15
```

- 配置接收进程

```
GGSCI> edit param rhxnl
```

```
EXTRACT rhxnl
```

```
SOURCEDEFS ./dirdef/ogg.def
```

```
CUSEREXIT ./flatfilewriter.so CUSEREXIT PASSTHRU INCLUDEUPDATEBEFORES,
```

```
PARAMS "/dirprm/rhxnl.properties"
```

```
TABLE ogg_test.*;
```

- 添加进程：ADD EXTRACT rhxnl, EXTTRAILSOURCE ./dirdat/no

- 把源端生成的 ogg.def 拷贝到目标端 dirdef 文件夹下

- 把目标端下这个文件/AdapterExamples/file-writer/ ffue.properties 拷贝到

目标端 `dirprm` 文件夹下,或者从别处复制一个 `.properties` 文件

- 启动 mgr: `start mgr`
- 启动进程: `start rhxnl`

当看到 mgr 和 rhxnl 状态都是 `running` 时,说明目标端配置完成,现在可以去调整源端进行测试

```
GGSCI (mochou) 1> info all
```

Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	RUNNING			
EXTRACT	RUNNING	FFWRITER	00:00:00	00:00:03

2.1.1.3.2.7.源端进程开启

```
GGSCI> start hxnl
```

```
GGSCI> start phxnl
```

通过 `info all` 查看进程状态,若进程显示 `running` 则正常运行,若是 `ABENDING` 或者 `STOPPED` 则代表启动错误

通过 `view report *` (*是进程名)来查看运行日志,空格到最后会有报错日志

```
2017-08-18 11:27:48 ERROR OGG-01044 The trail './dirdat/st' is not assigned to extract 'PUMP'. Assign the trail to the extract with the command "ADD EXTTRAIL/RMTTRAIL ./dirdat/st, EXTRACT PUMP".
```

这个报错已经说得很清楚了可以通过下面命令来解决

```
GGSCI>ADD EXTRAIL ./dirdat/st , extract pump
```

至于 `ERROR OGG-10668 ABENDING` 这个错误是代表着配置有错误,即只要有错误配置这个 `ERROR` 就会出现,所以不用管它,只需更改好正确配置,错误标志就会消失

在 `.properties` 文件里配置的一定要在目标端建立好,譬如 `log` 存放日志(`mkdir log`), `trail` 文件存放位置(一般在 `dirdat` 下,如果不建好文件夹,源端找不到目标文件夹,则 `trail` 文件无法传过来,源端投递进程会无法开启)以及生成的 `dsv` 文件夹存放目录(一般在 `dirout` 文件夹下)

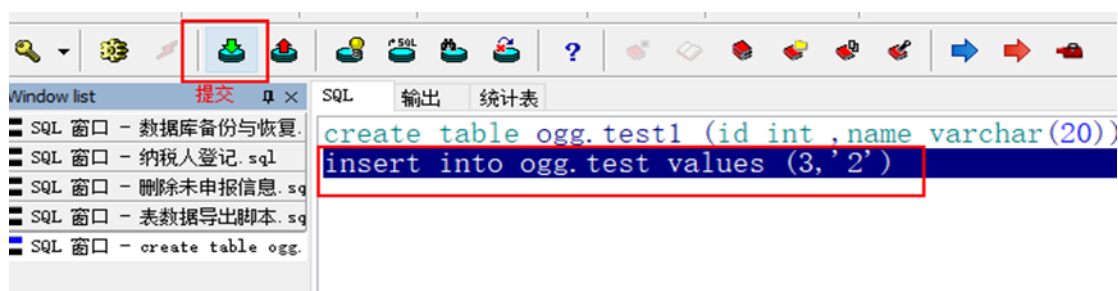
当配置的抽取和投递进程都显示 `RUNNING` 时,即代表源端,目标端都已准备好

```
GGSCI (mocho) 9> info all
```

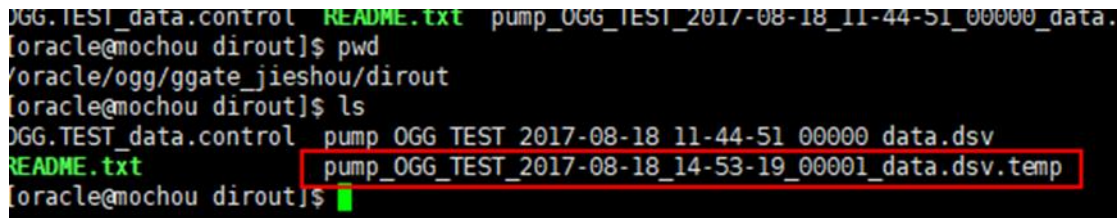
Program	Status	Group	Lag at Chkpt	Time Since Chkpt
MANAGER	RUNNING			
EXTRACT	ABENDED	EXT1	03:58:55	19:38:44
EXTRACT	RUNNING	EXTRACT	00:00:00	00:00:03
EXTRACT	RUNNING	PUMP	00:00:00	00:00:02
REPLICAT	STOPPED	REP1	00:00:00	19:31:31

2.1.1.3.3. 测试 OGG

- 在 pl/sql 里插入 ogg_test.test 表一行数据，并点击提交 commit



- 然后在目标端 dirout 文件夹下就可以看到有文件传进来，文件是由时间戳来命名的



- Strings/cat/more/vim 等命令查看该 dsv 文件，即可得到更改讯息事件 id，事件类型，发生时间，表，表数据等，其中事件类型有三类(I:insert, D:delete, U:update)，这个文件也可以通过 datax 配置 path 路径上云
- GoldenGate File Adapter 提供了两个文件，用来生成数据文件。其中 properties 中有多个文件格式控制属性，通过设置这些属性，即可以控制生成文件的格式，它的文件属性如下：

```
dsvwriter.includecolnames=false
```

是否在字段值前放置字段名称，缺省为 false

```
dsvwriter.files.onepertable=true
```

每个表格各生成一个文件，还是所有数据放入一个文件，缺省为 true

```
dsvwriter.files.data.rootdir=./dirout
```

数据文件输出目录

`dsvwriter.files.data.ext=_data.dsv`

已就绪的数据文件扩展名

`dsvwriter.files.data.tmpext=_data.dsv.temp`

处理中的数据文件

`dsvwriter.files.data.rollover.time=1800`

数据文件由“处理中”切换为“就绪”的最大时间（秒）

`dsvwriter.files.data.rollover.size=104857600`

数据文件由“处理中”切换为“就绪”的最大文件尺寸（KB）

`dsvwriter.files.data.norecords.timeout=1800`

当无记录写入时，最多等待时间（秒）即切换数据文件为“就绪”，缺省 120 秒

`dsvwriter.files.rolloveronshutdown=true`

当本属性为 `true` 时，如果 `Extract` 进程停止，则所有空白“处理中”文件被删除，所有有数据的“处理中”文件切换为“就绪”文件。当本属性为 `false` 时，如果 `Extract` 进程停止，则所有空白“处理中”文件被删除，所有有数据的“处理中”文件状态不变。

`dsvwriter.dsv.fielddelim.chars=|`

数据文件中的字段分隔符

`dsvwriter.dsv.linedelim.chars=\n`

数据文件中的行终结符号

`dsvwriter.dsv.quotes.chars="`

数据文件中的引号符号

`dsvwriter.dsv.quotes.escaped.chars=""`

数据文件中的 `escape` 符号

`dsvwriter.metacols=opcode,timestamp`

数据文件中每行数据之前的元数据：

`Opcode` — I, U 与 D 代表 Insert, Update 和 Delete `Timestamp` — 记录的提交时间戳

```
dsvwriter.metacols.opcode.insert.chars=I
```

数据文件中代表 Insert 操作的字符

```
dsvwriter.metacols.opcode.update.chars=U
```

数据文件中代表 Update 操作的字符

```
dsvwriter.metacols.opcode.delete.chars=D
```

数据文件中代表 Delete 操作的字符

```
dsvwriter.files.formatstring=pump_%s_%t_%d_%05n
```

数据文件名字格式: %s — schema

%t — table

%d — timestamp

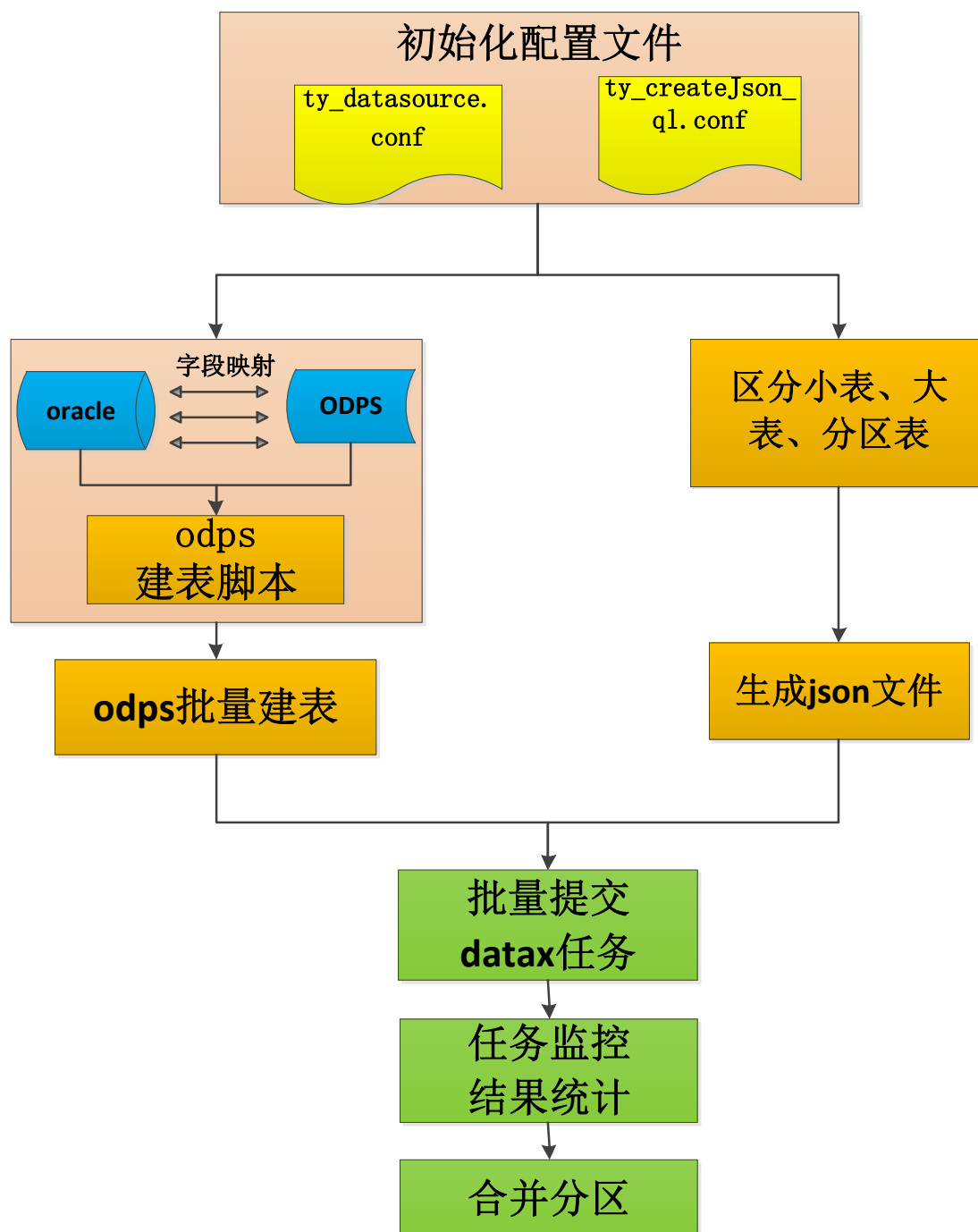
%05n — 5 位序号

2.2. Datax 全量初始化实施

2.2.1. 总体流程及结构

2.2.1.1. 流程图

首先配置好 `ty_datasource.conf` 和 `ty_createJson_ql.conf` 配置文件，分别通过这两个配置文件生成 `odps` 建表脚本和所有 `json` 文件，再在 `odps` 上创建所有表，再通过并发脚本提交 `datax` 任务提交，或通过定时任务提交，实现数据上云。整个流程图如下：



2.2.1.2. 文件目录结构

整个文件目录如下图

一级 TY: 层级:贴源 (其他层如标准 BZ)

二级: 1) 系统代码如核心征管为 HX

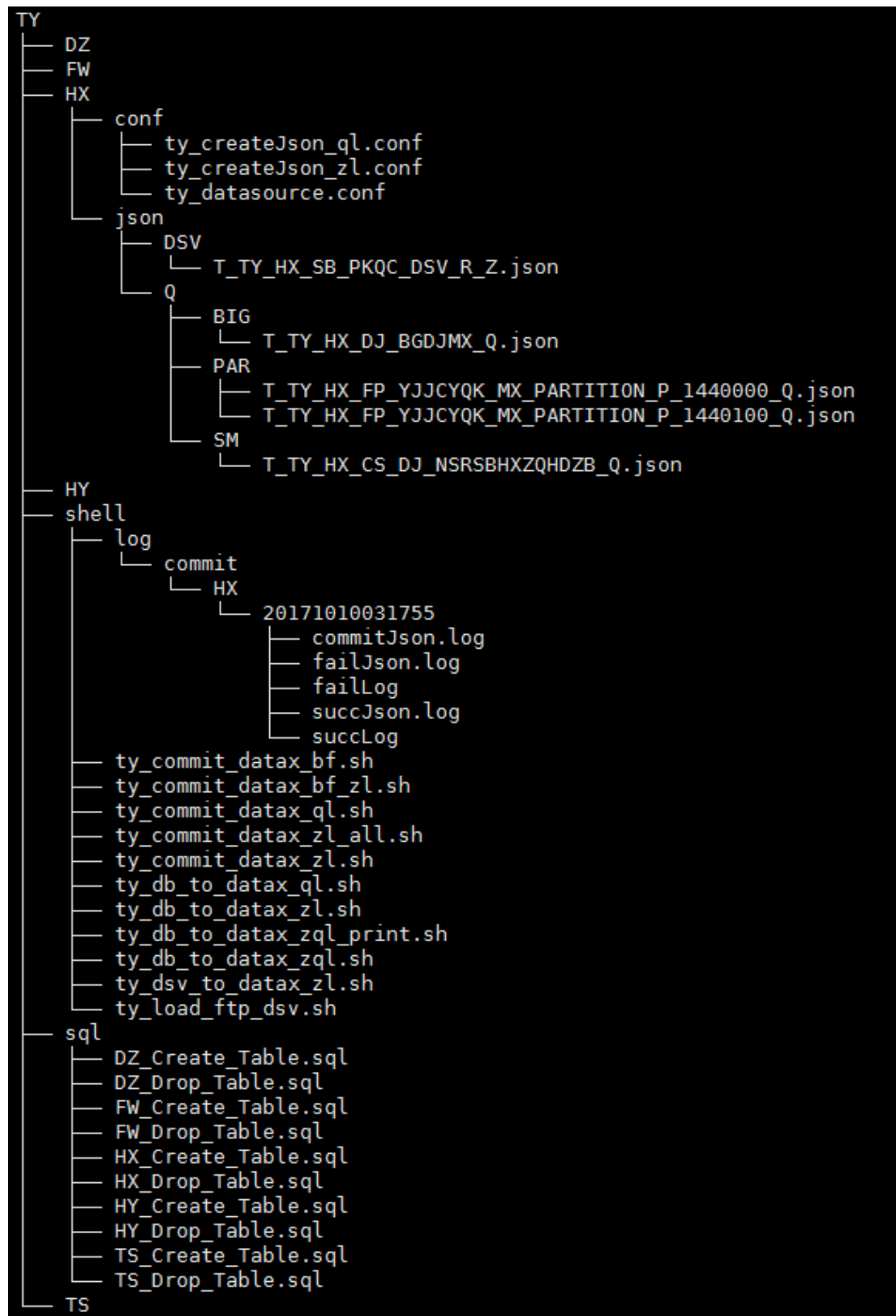
2) shell 所有的 shell 脚本目录

3) sql 存放 odps 建表语句及删表语句

三级：1) conf: 配置文件目录

2) json 存放所有的 json 文件，包括全量（Q）和增量（DSV）

3) log 提交任务产生的日志保存目录



2.2.2. 初始化配置

以下路径均以 HX 为例说明

2.2.2.1. 配置 ty_datasource.conf

目录为 TY/HX/conf

根据现场实际情况配置以下信息，主要包括 odps、oracle、ftp 服务器相关信息

```
#####
#
#ODPS 配置          --下面部分都是 ODPS 配置的内容
#####
#
odpsServer="http://service.odps.aliyun.com/api"          #ODPS 服务器地址
tunnelServer=""          #ODPS tunnel 地址(未使用)
odpsaccessId="*****"          #ODPS 的 accessid
odpsaccessKey="*****"          #ODPS 的 accesskey
project="SC_JC_TY"#项目名称
errorLimit=10#允许错误记录数(未使用)
dataxLog="/home/datax/log"#datax 日志目录(未使用)
#####
#
#源库配置
#####
#
#数据读取类型: oraclereader、mysqlreader
reader="oraclereader"
#reader="mysqlreader"
jdbc="xx.xx.xx.x:1521/xxxx"#连接源库 jdbc 地址
user="dsjsy"#源库用户名
pass="*****"#源库密码
odpsJdbc="jdbc:oracle:thin:@${jdbc}"#连接 odps 的 jdbc
#####
#
#文件服务器地址，用于存储增量 dsv 文件
#####
#
ftplp="xxx.xx.xx.xx"##ftp 服务器地址
ftpPort="22"##ftp 服务器地址
ftpUser="root"##ftp 服务器用户名
ftpPasswd="*****"##ftp 服务器密码
ftpBasePath="/home/DSV/HX"
```

```
#####
#
#其他配置
#####
#
##是否开启版本控制
checkVersion=false##开启版本控制(未使用)
##ogg 增量文件后缀
oggSubfix="dsv"
##ogg 增量文件分隔符
spiltFlag="@@"
##oracle_home 配置
TY_ORACLE_HOME="/oradata/app/oracle/product/11.2.0/db_1"
##oracle_lib 配置
TY_LD_LIBRARY_PATH="${TY_ORACLE_HOME}/lib"
#####
#
```

2.2.2.2. 配置 ty_createJson_ql.conf

目录为 TY/HX/conf

此文件需要根据数据字典手动配置，一张表对应一行数据

格式为：域名|表名|odps 表名|主键|是否为分区表|表记录数

```
#####域名|表名|odps 表名|主键|是否为分区表|表记录数
HX_CS_QG|CS_DJ_SXXDGDQGLB|T_TY_HX_CS_DJ_SXXDGDQGLB|SSXD_DM|NO|159
HX_CS_QG|CS_DJ_SWSZZJLXDZB|T_TY_HX_CS_DJ_SWSZZJLXDZB|SWZJZL_DM|NO|46
```

```
[root@centos1 conf]# more ty_createJson_ql.conf
#####域名|表名|odps表名|主键|是否为分区表|表记录数
HX_CS_QG|CS_DJ_SXXDGDQGLB|T_TY_HX_CS_DJ_SXXDGDQGLB|SSXD_DM|NO|159
HX_CS_QG|CS_DJ_SWSZZJLXDZB|T_TY_HX_CS_DJ_SWSZZJLXDZB|SWZJZL_DM|NO|46
HX_CS_QG|CS_DJ_YGZSDZZSPMYZSPMDZB|T_TY_HX_CS_DJ_YGZSDZZSPMYZSPMDZB|UUID|NO|84
HX_CS_QG|CS_DJ_ZDSYJKJCDZB|T_TY_HX_CS_DJ_ZDSYJKJCDZB|JGCJ_DM|NO|5
HX_CS_QG|CS_DJ_ZJZLDJZCLXDZB|T_TY_HX_CS_DJ_ZJZLDJZCLXDZB|SFZJZL_DM|NO|30
HX_CS_QG|CS_FP_FPDKKXZSPM|T_TY_HX_CS_FP_FPDKKXZSPM|ZSPM_DM|NO|10
HX_CS_QG|CS_FP_HWYSZPBKXDZSPM|T_TY_HX_CS_FP_HWYSZPBKXDZSPM|ZSXM_DM|NO|167
HX_CS_QG|CS_FP_YGZJYLBZSPMDZ|T_TY_HX_CS_FP_YGZJYLBZSPMDZ|ZSPM_DM|NO|187
HX_CS_QG|CS_GY_XJDMZDZB|T_TY_HX_CS_GY_XJDMZDZB|UUID|NO|189
HX_CS_QG|CS_JC_SSWFXWCLCFYJDZB|T_TY_HX_CS_JC_SSWFXWCLCFYJDZB|UUID|NO|3412
HX_CS_QG|CS_JC_WFSDYJCFGX|T_TY_HX_CS_JC_WFSDYJCFGX|WFWZXW_DM|NO|94
HX_CS_QG|CS_PZ_PZFLB|T_TY_HX_CS_PZ_PZFLB|PZFL_DM|NO|15
HX_CS_QG|CS_PZ_PZLFLGLB|T_TY_HX_CS_PZ_PZLFLGLB|PZLFLGLUUUID|NO|58
HX_CS_QG|CS_RD_SXXDSDLXSLDZB|T_TY_HX_CS_RD_SXXDSDLXSLDZB|SSXDSDLX_DM|NO|14
HX_CS_QG|CS_RD_XZQHXYXZDZB|T_TY_HX_CS_RD_XZQHXYXZDZB|XZQHSZ_DM|NO|56
HX_CS_QG|CS_RD_ZZSJYBFZSHWZSLDZB|T_TY_HX_CS_RD_ZZSJYBFZSHWZSLDZB|JYBFZSZSHWLX_DM|NO|48
HX_CS_QG|CS_SB_HSQJYSKMDZB|T_TY_HX_CS_SB_HSQJYSKMDZB|XH|NO|15
HX_CS_QG|CS_SB_HYHDLRLDZB|T_TY_HX_CS_SB_HYHDLRLDZB|HY_DM|NO|4
HX_CS_QG|CS_SB_KQSYFFL|T_TY_HX_CS_SB_KQSYFFL|ZSPM_DM|NO|20
HX_CS_QG|CS_SB_MSSR|T_TY_HX_CS_SB_MSSR|UUID|NO|7
HX_CS_QG|CS_SB_QSQSZYYZSPMDZB|T_TY_HX_CS_SB_QSQSZYYZSPMDZB|UUID|NO|115
HX_CS_QG|CS_SB_QYSDS_HYDZB|T_TY_HX_CS_SB_QYSDS_HYDZB|UUID|NO|100
```

2.2.2.3. odps 建表

2.2.2.3.1. 源表与目标表字段类型映射

根据 odps 所支持的字段类型，我们定义 oracle 与 odps 的字段类型的映射关系如下

oracle 数据类型 ^o	ODPS 数据类型 ^o
char ^o	String ^o
varchar ^o	String ^o
varchar2 ^o	String ^o
Date ^o	DateTime ^o
TIMESTAMP ^o	DateTime ^o
BINARY_FLOAT ^o	double ^o
BINARY_DOUBLE ^o	double ^o
FLOAT ^o	double ^o
integer ^o	bigint ^o
number(x) x<19 ^o	bigint ^o
number(x) x<36 ^o	decimal ^o
number(p,s) p-s 小于等于 36, s 小于等于 18, 例如 NUMBER(38,2) ^o	Decimal ^o
number 其他 ^o	String ^o
RAW ^o	String ^o
UROWID ^o	String ^o

这里 shell 脚本中已做了映射处理，此步骤无需操作。

2.2.2.3.2. 生成建表脚本

shell 目录下执行脚本：./ty_create_table_sql.sh HX

```
[root@dsjpt2 shell]# ./ty_create_table_sql.sh HX
数据库连接成功，开始执行脚本！
开始：CS_DJ_SXXDGJDQGLB---HX_CS_QG
【=====检测完毕=====】
【库表检测：HX_CS_QG.CS_DJ_SXXDGJDQGLB】
生成HX_CS_QG.CS_DJ_SXXDGJDQGLB建表语句
开始：CS_DJ_SWSZZJLXDZB---HX_CS_QG
【=====检测完毕=====】
【库表检测：HX_CS_QG.CS_DJ_SWSZZJLXDZB】
生成HX_CS_QG.CS_DJ_SWSZZJLXDZB建表语句
开始：CS_DJ_YGZSDZZSPMYZSPMDZB---HX_CS_QG
【=====检测完毕=====】
【库表检测：HX_CS_QG.CS_DJ_YGZSDZZSPMYZSPMDZB】
生成HX_CS_QG.CS_DJ_YGZSDZZSPMYZSPMDZB建表语句
开始：CS_DJ_ZDSYJKJCDZB---HX_CS_QG
【=====检测完毕=====】
【库表检测：HX_CS_QG.CS_DJ_ZDSYJKJCDZB】
生成HX_CS_QG.CS_DJ_ZDSYJKJCDZB建表语句
开始：CS_DJ_ZJZLDJZCLXDZB---HX_CS_QG
【=====检测完毕=====】
【库表检测：HX_CS_QG.CS_DJ_ZJZLDJZCLXDZB】
生成HX_CS_QG.CS_DJ_ZJZLDJZCLXDZB建表语句
```

执行完成后，在 TY/sql 下生成 HX_Create_Table.sql 文件

```
[root@centos1 sql]# pwd
/home/admin/version/TY/sql
[root@centos1 sql]# ll
total 3724
-rw-r--r-- 1 root root 126259 Oct 23 07:58 DZ_Create_Table.sql
-rw-r--r-- 1 root root 2870 Oct 23 07:58 DZ_Drop_Table.sql
-rw-r--r-- 1 root root 88185 Oct 23 07:58 FW_Create_Table.sql
-rw-r--r-- 1 root root 3977 Oct 23 07:58 FW_Drop_Table.sql
-rw-r--r-- 1 root root 2314931 Oct 23 08:29 HX_Create_Table.sql
-rw-r--r-- 1 root root 98574 Oct 23 07:58 HX_Drop_Table.sql
-rw-r--r-- 1 root root 57158 Oct 23 07:58 HY_Create_Table.sql
-rw-r--r-- 1 root root 4736 Oct 23 07:58 HY_Drop_Table.sql
-rw-r--r-- 1 root root 1081005 Oct 23 07:58 TS_Create_Table.sql
-rw-r--r-- 1 root root 17522 Oct 23 07:58 TS_Drop_Table.sql
[root@centos1 sql]#
```

在源表基础上新增了以下字段：

ypt_jgsj（云平台数据加工时间）

ypt_ysjczlx（云平台源数据操作类型）

ypt_ysjczsj（云平台源数据操作时间）

ypt_ysjczxl（云平台源数据操作序列）

```
create table if not exists t_ty_hx_cs_dj_sxxdgjdqglb
(ssxd_dm string comment '税收协定代码',
gjhdqsz_dm string comment '国家或地区数字代码',
yxbz string comment '有效标志',
xybz string comment '选用标志',
ypt_jgsj datetime comment '云平台数据加工时间',
ypt_ysjczlx string comment '源数据操作类型',
ypt_ysjczsj datetime comment '源数据操作时间',
ypt_ysjczxl string comment '源数据操作序列')
comment '税收协定国家地区关联表'
partitioned by (rfq string comment '日分区');
create table if not exists t_ty_hx_cs_dj_swszzjlxdb
(swszzl_dm string comment '税务证件种类代码',
yxbz string comment '选用标志',
yxbz string comment '有效标志',
lcswsx_dm string comment '流程税务事项代码',
dzbzdszl_dm string comment '电子表证单书种类代码',
ypt_jgsj datetime comment '云平台数据加工时间',
ypt_ysjczlx string comment '源数据操作类型',
ypt_ysjczsj datetime comment '源数据操作时间',
ypt_ysjczxl string comment '源数据操作序列')
comment '税务事项证件类型对照表'
partitioned by (rfq string comment '日分区');
create table if not exists t_ty_hx_cs_dj_ygzsdzzspmzspmdzb
(uuid string comment 'uuid||uuid',
ygzsdzzspm_dm string comment '营改增试点主征收品目代码',
zspm_dm string comment '征收品目代码',
yxbz string comment '选用标志',
yxbz string comment '有效标志',
ypt_jgsj datetime comment '云平台数据加工时间',
ypt_ysjczlx string comment '源数据操作类型',
ypt_ysjczsj datetime comment '源数据操作时间',
ypt_ysjczxl string comment '源数据操作序列')
comment '营改增试点主征收品目与征收品目对照表'
partitioned by (rfq string comment '日分区');
create table if not exists t_ty_hx_cs_dj_zdsyjkjcdzb
(jgcj_dm string comment '机关层级代码',
zdsyhjkjc_dm string comment '重点税源户监控级次代码',
```

2.2.2.3.3. odps 建表

在 odps 客户端（如未安装请先安装，并配置好 odps_config.ini）bin 目录下执行以下命令

```
./odpscmd -f HX_Create_Table.sql #要指定文件所在路径
```

```
[root@dsjpt2 bin]# pwd
/home/sc_odps/odpscmd_public/bin
[root@dsjpt2 bin]# ./odpscmd -f /home/admin/version/TY/sql/HX_Create_Table.sql
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
```

2.2.2.4. 生成 Json 文件

Json 文件是指提交 datax 任务所对应的 json 文件

shell 目录下执行脚本

./ty_db_to_datax_ql.sh HX

```
[root@dsjpt2 shell]# pwd
/home/admin/version/TY/shell
[root@dsjpt2 shell]# ./ty_db_to_datax_ql.sh HX
```

执行完成后在 HX/json 目录下生成 Q（全量），其中 DSV 为增量目录

```
[root@centos1 json]# ll
total 8
drwxr-xr-x 2 root root 4096 Oct 10 04:00 DSV
drwxr-xr-x 5 root root 4096 Oct 23 06:15 Q
[root@centos1 json]#
```

全量目录下有 BIG、PAR、SM，分别表示大表，分区表、小表

```
[root@centos1 Q]# ll
total 228
drwxr-xr-x 2 root root 12288 Sep 16 11:11 BIG
drwxr-xr-x 2 root root 102400 Sep 16 11:11 PAR
drwxr-xr-x 2 root root 118784 Oct 23 06:16 SM
[root@centos1 Q]#
```

Json 文件示例：


```
{
  "job": {
    "content": [
      {
        "reader": {
          "name": "oraclereader",
          "parameter": {
            "column": ["HDQCUUID", "DJXH", "FPZL_DM", "DFFPZGKPXE_DM", "MYZGGPSL", "MCZGGPSL", "CPZGSL", "FPGPFS_DM", "YXQQ", "YXQZ", "YXBZ", "SWJG_DM", "DFFPZGKPXE", "WTDKBZ", "LRR_DM", "LRRQ", "XGR_DM", "XGRQ", "SJGSDQ", "SJTB_SJ", "LXKPSX", "LXKPLJXE", "'S'", "'00000'"],
            "splitPk": "HDQCUUID",
            "connection": [
              {
                "jdbcUrl": ["$odpsJdbc"],
                "table": ["HX_FP_PZHDXX"]
              }
            ],
            "fetchSize": 1024,
            "password": "$pass",
            "username": "$user",
            "mandatoryEncoding": "UTF-8"
          }
        },
        "writer": {
          "name": "odpswriter",
          "parameter": {
            "accessId": "$odpsaccessId",
            "accessKey": "$odpsaccessKey",
            "accountType": "aliyun",
            "column": ["HDQCUUID", "DJXH", "FPZL_DM", "DFFPZGKPXE_DM", "MYZGGPSL", "MCZGGPSL", "CPZGSL", "FPGPFS_DM", "YXQQ", "YXQZ", "YXBZ", "DM", "DFFPZGKPXE", "WTDKBZ", "LRR_DM", "LRRQ", "XGR_DM", "XGRQ", "SJGSDQ", "SJTB_SJ", "LXKPSX", "LXKPLJXE", "ypt_ysjczlx", "ypt_ysjczxl"],
            "odpsServer": "http://service.cn-guangzhou-gdgs-d01.odps.data.gdgs.sat.tax/api",
            "partition": "rfq=$bizdate",
            "project": "$project",
            "table": "I_TY_HX_FP_PZHDXX",
            "truncate": "true"
          }
        }
      }
    ]
  }
}
```

2.2.3. 提交任务上云

2.2.3.1. 统计源表记录数

提交任务之前需要提交统计源表记录数，此步骤是为了统计结果用，如不需要统计，请跳过
shell/tools 目录下执行脚本，执行需要一段时间，请耐心等待

./ty_db_account.sh HX

```
[root@dsjpt2 tools]# pwd
/home/admin/version/TY/shell/tools
[root@dsjpt2 tools]# ./ty_db_account.sh HX
数据库连接成功，开始执行脚本！
统计成功，记录文件：shell/log/count/HX_CountTable.conf
run is success!
[root@dsjpt2 tools]#
```

2.2.3.2. 提交 datax 任务

shell 目录下执行脚本，

./ty_commit_datax_bf.sh HX 10 20171104 2>&1 >/dev/null &

```
[root@dsjpt2 shell]# pwd
/home/admin/version/TY/shell
[root@dsjpt2 shell]# ./ty_commit_datax_bf.sh HX 10 20171104 2>&1 >/dev/null &
```

建议通过配置 crontab 表达式定时在凌晨执行

```
0 0 5 11 * sh /home/admin/version/TY/shell/ty_commit_datax_bf.sh HX 10 20171104
2>&1 >/dev/null &
```

其中参数 10 为并发数任务数，可根据机器硬件配置适当调整,20171104 为全量分区

任务提交后可通过查看日志来监控任务的运行情况，日志目录在 TY/shell/log/commit/HX 下，以脚本开始运行的时间戳为名称创建文件夹

```
[root@dsjpt2 20171105000001]# pwd
/home/admin/version/TY/shell/log/commit/HX/20171105000001
[root@dsjpt2 20171105000001]# ll
total 804
-rw-r--r-- 1 root root 242242 Nov  5 17:32 commitJson.log
-rw-r--r-- 1 root root    30 Nov  5 13:51 failJson.log
drwxr-xr-x 2 root root  4096 Nov  5 13:51 failLog
-rw-r--r-- 1 root root 106631 Nov  5 17:32 succJson.log
drwxr-xr-x 2 root root 225280 Nov  9 10:29 succLog
-rw-r--r-- 1 root root  9766 Nov  5 10:49 T_TY_HX_SB_ZLBS_QYTQZLDJ_Q.log
-rw-r--r-- 1 root root 10937 Nov  5 10:51 T_TY_HX_SB_ZLBS_XQYKJZZFZ_Q.log
-rw-r--r-- 1 root root 10685 Nov  5 10:50 T_TY_HX_SB_ZLBS_ZLFXSXXB_MX_Q.log
-rw-r--r-- 1 root root 86486 Nov  5 11:06 T_TY_HX_SB_ZQJMFQ_Q.log
-rw-r--r-- 1 root root 16225 Nov  5 16:22 T_TY_HX_ZS_SKY_SFXQ_Q.log
-rw-r--r-- 1 root root 12488 Nov  5 16:22 T_TY_HX_ZS_TDSF_SQ_Q.log
-rw-r--r-- 1 root root 15444 Nov  5 16:34 T_TY_HX_ZS_YDTXX_Q.log
-rw-r--r-- 1 root root 10281 Nov  5 16:45 T_TY_HX_ZS_YQJNSK_SQB_Q.log
-rw-r--r-- 1 root root  9421 Nov  5 16:45 T_TY_HX_ZS_YQJNSK_SQMXB_Q.log
-rw-r--r-- 1 root root 15993 Nov  5 16:48 T_TY_HX_ZS_ZKSSWSZM_Q.log
[root@dsjpt2 20171105000001]#
```

可以看到以下任务指标

1、已运行的任务（包括正在运行的）：通过查看 commitJson.log 文件提交总的 json 数

```
[root@dsjpt2 20171105000001]# more commitJson.log
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_DJ_SSXDGDQGLB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_DJ_SWSZZJLXDZB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_DJ_YGZSDZZSPMYZSPMDZB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_DJ_ZDSYJKJCDZB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_DJ_ZJZLDJZCLXDZB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_FP_FPDKKXZSPM_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_FP_HWYSZPBKXDZSPM_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_FP_YGZJYLBZSPMDZ_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_GY_XJDMZDZB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_JC_SSWFXWCLCFYJDZB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_JC_WFSDYJCFGX_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_PZ_PZFLB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_PZ_PZZLFLGLB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_RD_SSXDSDLXSLDZB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_RD_XZQHXYXZDZB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_RD_ZZSJYBFZSHWZSLDZB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_SB_HSQJYSKMDZB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_SB_HYHDLRLDZB_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_SB_KQSYFFL_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_CS_SB_MSSR_Q.json
```

2、运行中：当前目录下的日志文件即为运行中的任务，运行中的任务数量与并发数相同，由图中可以看到正在运行任务数为 10

3、成功的任务：在 succLog 目录下为已运行成功的任务

```
[root@dsjpt2 succLog]# ll
total 138060
-rw-r--r-- 1 root root 9089 Nov 5 00:02 T_TY_HX_CS_DJ_JDXZXZQHDZB_Q.log
-rw-r--r-- 1 root root 8809 Nov 5 00:02 T_TY_HX_CS_DJ_NSRSBHXZQHDZB_Q.log
-rw-r--r-- 1 root root 9189 Nov 5 16:46 T_TY_HX_CS_DJ_QYHXPZB_Q.log
-rw-r--r-- 1 root root 9002 Nov 5 00:00 T_TY_HX_CS_DJ_SSXDGDQGLB_Q.log
-rw-r--r-- 1 root root 8793 Nov 5 00:02 T_TY_HX_CS_DJ_SWJGGXGWXZQHDZB_Q.log
-rw-r--r-- 1 root root 9096 Nov 5 00:00 T_TY_HX_CS_DJ_SWSZZJLXDZB_Q.log
-rw-r--r-- 1 root root 8813 Nov 5 00:02 T_TY_HX_CS_DJ_TDDJZSPMDZB_Q.log
-rw-r--r-- 1 root root 9010 Nov 5 00:02 T_TY_HX_CS_DJ_TDDWSESZB_Q.log
-rw-r--r-- 1 root root 8706 Nov 5 00:02 T_TY_HX_CS_DJ_XMDJSSJXSZ_Q.log
-rw-r--r-- 1 root root 9215 Nov 5 00:03 T_TY_HX_CS_DJ_XMDJZGSWSFPSZB_Q.log
-rw-r--r-- 1 root root 9117 Nov 5 00:00 T_TY_HX_CS_DJ_YGZSDZZSPMYZSPMDZB_Q.log
-rw-r--r-- 1 root root 9487 Nov 5 00:03 T_TY_HX_CS_DJ_YGZZHYZZSPMDZB_Q.log
-rw-r--r-- 1 root root 9000 Nov 5 00:00 T_TY_HX_CS_DJ_ZDSYKJCDZB_Q.log
-rw-r--r-- 1 root root 9018 Nov 5 00:00 T_TY_HX_CS_DJ_ZJZLDJZCLXDZB_Q.log
-rw-r--r-- 1 root root 9403 Nov 5 00:03 T_TY_HX_CS_DJ_ZXQYDJSKJKDZB_Q.log
-rw-r--r-- 1 root root 9659 Nov 5 00:03 T_TY_HX_CS_DJ_ZXQYSKJGDZB_Q.log
-rw-r--r-- 1 root root 9106 Nov 5 00:00 T_TY_HX_CS_FP_FPDKKXZSPM_Q.log
```

具体会记录任务的读出记录总数、读写失败总数、任务平均流量、任务总计耗时等信息

```
2017-11-05 00:07:25.830 [0-0-0-writer] INFO OdpsWriter$Task - Slave which uploadId=[2017110500063329b51056004405e8] commit blocks 0
2017-11-05 00:07:25.897 [taskGroup-0] INFO TaskGroupContainer - taskGroup[0] taskId[0] is succeeded, used[77028]ms
2017-11-05 00:07:25.898 [taskGroup-0] INFO TaskGroupContainer - taskGroup[0] completed it's tasks.
2017-11-05 00:07:32.168 [job-0] INFO StandAloneJobContainerCommunicator - Total 97459 records, 10400057 bytes | Speed 863.14KB/s, 8
ords/s | Error 0 records, 0 bytes | All Task WaitWriterTime 18.221s | All Task WaitReaderTime 18.079s | Percentage 100.00%
2017-11-05 00:07:32.169 [job-0] INFO AbstractScheduler - Scheduler accomplished all tasks.
2017-11-05 00:07:32.170 [job-0] INFO JobContainer - DataX Writer.Job [odpswriter] do post work.
2017-11-05 00:07:32.170 [job-0] INFO JobContainer - DataX Reader.Job [oraclereader] do post work.
2017-11-05 00:07:32.170 [job-0] INFO JobContainer - DataX jobId [0] completed successfully.
2017-11-05 00:07:32.172 [job-0] INFO HookInvoker - No hook invoked, because base dir not exists or is a file: /home/admin/datax3/ho
2017-11-05 00:07:32.175 [job-0] INFO JobContainer -
[total cpu info] =>
averageCpu | maxDeltaCpu | minDeltaCpu
1.29% | 1.29% | 1.29%

[total gc info] =>
NAME | totalGCCount | maxDeltaGCCount | minDeltaGCCount | totalGCTime | maxDelta
| minDeltaGCTime
| 0.000s PS MarkSweep | 0 | 0 | 0 | 0.000s | 0.000s
| 3.538s PS Scavenge | 5 | 5 | 5 | 3.538s | 3.538s

2017-11-05 00:07:32.175 [job-0] INFO JobContainer - PerfTrace not enable!
2017-11-05 00:07:32.176 [job-0] INFO StandAloneJobContainerCommunicator - Total 97459 records, 10400057 bytes | Speed 119.49KB/s, 1
ords/s | Error 0 records, 0 bytes | All Task WaitWriterTime 18.221s | All Task WaitReaderTime 18.079s | Percentage 100.00%
2017-11-05 00:07:32.178 [job-0] INFO JobContainer -
任务启动时刻 : 2017-11-05 00:05:29
任务结束时刻 : 2017-11-05 00:07:32
任务总计耗时 : 122s
任务平均流量 : 119.49KB/s
记录写入速度 : 1146rec/s
读出记录总数 : 97459
读写失败总数 : 0
[root@dsjpt2 succLog]#
```

4、失败的任务：在 failLog 目录下为已运行失败的任务

```
[root@dsjpt2 failLog]# ll
total 1076
-rw-r--r-- 1 root root 1094689 Nov 5 13:51 T_TY_HX_SB_CWBB_XQYKJZZ_LRB_Q.log
[root@dsjpt2 failLog]#
```

可查看具体的错误日志信息

```
at com.alibaba.datax.plugin.reader.oraclereader.OracleReader$Task.startRead(OracleReader.java:110) ~[oraclereader-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.taskgroup.runner.ReaderRunner.run(ReaderRunner.java:57) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at java.lang.Thread.run(Thread.java:744) [na:1.7.0_45]
2017-11-05 13:51:08.170 [job-0] INFO StandAloneJobContainerCommunicator - Total 346379892 records, 47725265945 bytes | Speed 82.57K
4 records/s | Error 0 records, 0 bytes | All Task WaitWriterTime 17,893.109s | All Task WaitReaderTime 145,344.141s | Percentage 9
2017-11-05 13:51:08.170 [job-0] ERROR JobContainer - 运行scheduler 模式[standalone]出错.
2017-11-05 13:51:08.171 [job-0] ERROR JobContainer - Exception when job run
com.alibaba.datax.common.exception.DataXException: Code:[DBUtilErrorCode-07], Description:[读取数据库数据失败. 请检查您的配置的 column/where/querySql或者向 DBA 寻求帮助.]. - 执行的SQL为: select UUID,ZLBSCJUUUID,EWBHXH,HMC,BYJE,BNLJJE,LRR_DM,LRRQ,XGRQ,XGR_DM,SJGSDQ,
,'S','000000' from HX_SB.SB_CWBB_XQYKJZZ_LRB where ('588DE9B3B9A208DAECFDEE958A381979' <= UUID AND UUID < '7D5B1552E10D6835A13822A3
7') 具体错误信息为: java.sql.SQLException: ORA-01555: snapshot too old: rollback segment number 30 with name "_SYSSMU30_286768694$"
all

at com.alibaba.datax.common.exception.DataXException.asDataXException(DataXException.java:26) ~[datax-common-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.plugin.rdbms.util.RdbmsException.asQueryException(RdbmsException.java:93) ~[na:na]
at com.alibaba.datax.plugin.rdbms.reader.CommonRdbmsReader$Task.startRead(CommonRdbmsReader.java:220) ~[na:na]
at com.alibaba.datax.plugin.reader.oraclereader.OracleReader$Task.startRead(OracleReader.java:110) ~[na:na]
at com.alibaba.datax.core.taskgroup.runner.ReaderRunner.run(ReaderRunner.java:57) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at java.lang.Thread.run(Thread.java:744) ~[na:1.7.0_45]
2017-11-05 13:51:08.172 [job-0] INFO StandAloneJobContainerCommunicator - Total 346379892 records, 47725265945 bytes | Speed 44.456
6379892 records/s | Error 0 records, 0 bytes | All Task WaitWriterTime 17,893.109s | All Task WaitReaderTime 145,344.141s | Perce
.04%
2017-11-05 13:51:08.173 [job-0] ERROR Engine -

经DataX智能分析,该任务最可能的错误原因是:
com.alibaba.datax.common.exception.DataXException: Code:[DBUtilErrorCode-07], Description:[读取数据库数据失败. 请检查您的配置的 column/where/querySql或者向 DBA 寻求帮助.]. - 执行的SQL为: select UUID,ZLBSCJUUUID,EWBHXH,HMC,BYJE,BNLJJE,LRR_DM,LRRQ,XGRQ,XGR_DM,SJGSDQ,
,'S','000000' from HX_SB.SB_CWBB_XQYKJZZ_LRB where ('588DE9B3B9A208DAECFDEE958A381979' <= UUID AND UUID < '7D5B1552E10D6835A13822A3
7') 具体错误信息为: java.sql.SQLException: ORA-01555: snapshot too old: rollback segment number 30 with name "_SYSSMU30_286768694$"
all

at com.alibaba.datax.common.exception.DataXException.asDataXException(DataXException.java:26)
at com.alibaba.datax.plugin.rdbms.util.RdbmsException.asQueryException(RdbmsException.java:93)
at com.alibaba.datax.plugin.rdbms.reader.CommonRdbmsReader$Task.startRead(CommonRdbmsReader.java:220)
at com.alibaba.datax.plugin.reader.oraclereader.OracleReader$Task.startRead(OracleReader.java:110)
at com.alibaba.datax.core.taskgroup.runner.ReaderRunner.run(ReaderRunner.java:57)
at java.lang.Thread.run(Thread.java:744)
```

5、总时长

所有任务运行完成后,可查看 commitJson.log,在最后记录了开始时间与结束时间,即可得到所有任务运行的总时长。

```
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_DM_SB_GZLX_Q.json
/home/admin/version/TY/HX/json/Q/SM/T_TY_HX_DM_GY_XTBM_Q.json
开始时间:20171105000001
结束时间:20171105173239
[root@dsjpt2 20171105000001]#
```

2.2.4.结果统计(非必须)

根据产生的日志信息可统计任务完成情况

shell/tools 下执行脚本

./ty_datax_result_ql.sh HX 日志目录的时间戳

```
[root@dsjpt2 tools]# pwd
/home/admin/version/TY/shell/tools
[root@dsjpt2 tools]# ./ty_datax_result_ql.sh HX 20171105000001
HX_CS_ZDY.CS_DJ_JDXZXZQHDZB succ
HX_CS_ZDY.CS_DJ_NSRSBHXZQHDZB succ
HX_CS_QG.CS_DJ_QYHXPZB succ
HX_CS_QG.CS_DJ_SXDXGJDQGLB succ
HX_CS_ZDY.CS_DJ_SWJGGXGWXZQHDZB succ
HX_CS_QG.CS_DJ_SWSZZJLXDZB succ
HX_CS_ZDY.CS_DJ_TDDJZSPMDZB succ
HX_CS_ZDY.CS_DJ_TDDWSESZB succ
HX_CS_ZDY.CS_DJ_XMDJSSJXSZ succ
HX_CS_ZDY.CS_DJ_XMDJZGSWSFPSZB succ
HX_CS_QG.CS_DJ_YGZSDZZSPMYZSPMDZB succ
HX_CS_ZDY.CS_DJ_YGZZHYZZSPMDZB succ
HX_CS_QG.CS_DJ_ZDSYJKJCDZB succ
HX_CS_QG.CS_DJ_ZJZLDJZCLXDZB succ
HX_CS_ZDY.CS_DJ_ZXQYDJSKJKDZB succ
HX_CS_ZDY.CS_DJ_ZXQYSKJGDZB succ
HX_CS_QG.CS_FP_FPDKKXZSPM succ
HX_CS_ZDY.CS_FP_FPDKSZDZB succ
HX_CS_ZDY.CS_FP_GSDZDSFJSDZB succ
HX_CS_QG.CS_FP_HWYSZPBKXDZSPM succ
HX_CS_QG.CS_FP_YGZJYLBYZSPMDZ succ
HX_CS_ZDY.CS_GY_GLB_ZSPM succ
HX_CS_ZDY.CS_GY_GLB_ZSXM succ
HX_CS_ZDY.CS_GY_GLB_ZSZM succ
HX_CS_ZDY.CS_GY_JGCJJGJCDZB succ
HX_CS_ZDY.CS_GY_JJR succ
```

统计结果在目录 TY/shell/log/count 下
以脚本运行的当天日期为文件夹名称

```
[root@dsjpt2 count]# ll
total 108
drwxr-xr-x 6 root root 4096 Sep 26 09:55 20170926
drwxr-xr-x 3 root root 4096 Oct 21 11:18 20171021
drwxr-xr-x 3 root root 4096 Oct 25 10:49 20171025
drwxr-xr-x 5 root root 4096 Oct 26 16:07 20171026
drwxr-xr-x 3 root root 4096 Oct 27 10:20 20171027
drwxr-xr-x 3 root root 4096 Nov 7 20:32 20171107
drwxr-xr-x 3 root root 4096 Nov 25 16:01 20171125
drwxr-xr-x 2 root root 4096 Nov 25 14:39 bak
-rw-r--r-- 1 root root 1738 Nov 25 15:15 DZ_CountTable.conf
-rw-r--r-- 1 root root 2216 Nov 2 20:26 FW_CountTable.conf
-rw-r--r-- 1 root root 46385 Nov 25 14:30 HX_CountTable.conf
-rw-r--r-- 1 root root 2838 Nov 16 16:12 HY_CountTable.conf
-rw-r--r-- 1 root root 9434 Oct 26 08:46 TS_CountTable.conf
```



```
[root@dsjpt2 count]# cd 20171125
[root@dsjpt2 20171125]# ll
total 12
drwxr-xr-x 2 root root 4096 Nov 25 16:01 HX_FailLog
-rw-r--r-- 1 root root    0 Nov 25 16:01 HX_FailResult.log
-rw-r--r-- 1 root root 4624 Nov 25 16:01 HX_SuccResult.log
[root@dsjpt2 20171125]#
```

```
[root@dsjpt2 20171125]# more HX_SuccResult.log
表名 启动时刻 结束时刻 耗时 平均流量 写入速度 源库表记录数 读出记录数 读写失败数
HX_CS_ZDY_CS_DJ_JDXZQZQDZB 2017-11-05 00:01:46 2017-11-05 00:02:20 34s 2.12KB/s 90rec/s 1809 1809 0
HX_CS_ZDY_CS_DJ_NSRBHXZQDZB 2017-11-05 00:02:02 2017-11-05 00:02:24 22s 6B/s 0rec/s 3 3 0
HX_CS_QG_CS_DJ_QYHXPZB 2017-11-05 16:45:54 2017-11-05 16:46:09 15s 73B/s 2rec/s 21 0 0
HX_CS_QG_CS_DJ_SSDXGJDQGLB 2017-11-05 00:00:03 2017-11-05 00:00:21 18s 254B/s 15rec/s 159 159 0
HX_CS_ZDY_CS_DJ_SWJGGXGWZQDZB 2017-11-05 00:02:08 2017-11-05 00:02:30 22s 452B/s 18rec/s 181 181 0
HX_CS_QG_CS_DJ_SWSZJLXDZB 2017-11-05 00:00:03 2017-11-05 00:00:22 19s 156B/s 4rec/s 46 46 0
HX_CS_ZDY_CS_DJ_TDDJZSPMDZB 2017-11-05 00:02:19 2017-11-05 00:02:38 19s 136B/s 4rec/s 44 44 0
HX_CS_ZDY_CS_DJ_TDDWSESZB 2017-11-05 00:02:21 2017-11-05 00:02:43 21s 5.68KB/s 87rec/s 873 873 0
HX_CS_ZDY_CS_DJ_XMDJSSJXSZ 2017-11-05 00:02:24 2017-11-05 00:02:43 19s 6B/s 0rec/s 3 3 0
HX_CS_ZDY_CS_DJ_XMDJZGSWSFSPZB 2017-11-05 00:02:29 2017-11-05 00:03:35 65s 0B/s 0rec/s 1 1 0
HX_CS_QG_CS_DJ_YGSDZSZSPMYZSPMDZB 2017-11-05 00:00:03 2017-11-05 00:00:21 18s 462B/s 8rec/s 84 84 0
HX_CS_ZDY_CS_DJ_YGZHZYZSPMDZB 2017-11-05 00:02:29 2017-11-05 00:03:19 50s 825B/s 8rec/s 165 165 0
HX_CS_QG_CS_DJ_ZDSYJKJCDZB 2017-11-05 00:00:03 2017-11-05 00:00:21 18s 5B/s 0rec/s 5 5 0
HX_CS_QG_CS_DJ_ZJLDJZCLXDZB 2017-11-05 00:00:03 2017-11-05 00:00:23 20s 42B/s 3rec/s 30 30 0
HX_CS_ZDY_CS_DJ_ZXQYDJSKJXDZB 2017-11-05 00:02:34 2017-11-05 00:03:28 54s 778B/s 12rec/s 420 420 0
HX_CS_ZDY_CS_DJ_ZXQYKJGDZB 2017-11-05 00:02:35 2017-11-05 00:03:30 54s 13B/s 0rec/s 5 5 0
HX_CS_QG_CS_FP_FPDKXZSPM 2017-11-05 00:00:03 2017-11-05 00:00:21 18s 29B/s 1rec/s 10 10 0
HX_CS_ZDY_CS_FP_FPDKSZDZB 2017-11-05 00:02:35 2017-11-05 00:03:14 39s 49B/s 2rec/s 41 41 0
HX_CS_ZDY_CS_FP_GSDZDZJSDZB 2017-11-05 00:02:34 2017-11-05 00:03:05 31s 49B/s 0rec/s 5 5 0
HX_CS_QG_CS_FP_HWYSZPBKXDZSPM 2017-11-05 00:00:03 2017-11-05 00:00:18 15s 498B/s 16rec/s 167 167 0
HX_CS_QG_CS_FP_YGZJYLBZSPMDZ 2017-11-05 00:00:03 2017-11-05 00:00:22 19s 563B/s 18rec/s 187 187 0
HX_CS_ZDY_CS_GY_GLB_ZSPM 2017-11-05 00:02:43 2017-11-05 00:03:23 39s 8.28KB/s 78rec/s 780 780 0
HX_CS_ZDY_CS_GY_GLB_ZSXM 2017-11-05 00:02:48 2017-11-05 00:03:39 50s 140B/s 3rec/s 35 35 0
HX_CS_ZDY_CS_GY_GLB_ZSZM 2017-11-05 00:02:49 2017-11-05 00:04:10 81s 688B/s 6rec/s 189 189 0
HX_CS_ZDY_CS_GY_JGCGJGCDZB 2017-11-05 00:02:59 2017-11-05 00:04:10 70s 24B/s 1rec/s 23 23 0
HX_CS_ZDY_CS_GY_JJR 2017-11-05 00:03:43 2017-11-05 00:04:10 27s 4.65KB/s 58rec/s 587 587 0
HX_CS_ZDY_CS_GY_KPXMYJZSFDZB 2017-11-05 00:03:34 2017-11-05 00:04:13 38s 47B/s 0rec/s 11 11 0
HX_CS_ZDY_CS_GY_SSJMXZJMFDEDSLX 2017-11-05 00:03:39 2017-11-05 00:04:26 46s 26B/s 0rec/s 5 5 0
HX_CS_ZDY_CS_GY_SSJMXZJMLZXDZB 2017-11-05 00:03:42 2017-11-05 00:04:12 30s 3.67KB/s 43rec/s 864 864 0
HX_CS_ZDY_CS_GY_SSJMXZSZSPMDZB 2017-11-05 00:03:34 2017-11-05 00:04:36 61s 3.28KB/s 37rec/s 755 755 0
HX_CS_ZDY_CS_GY_TSNJGDZ 2017-11-05 00:03:34 2017-11-05 00:04:45 71s 2.01KB/s 21rec/s 650 650 0
HX_CS_QG_CS_GY_XJDMZDZB 2017-11-05 00:00:04 2017-11-05 00:00:20 16s 1.68KB/s 18rec/s 189 189 0
```

2.2.5.合并分区

由于分区表是按源表的分区提交，即源表是什么分区，提交到 odps 就是哪个分区，所以需要多个分区合并到一个日分区。

1、配置分区表，在 TY/HX/json/Q/PAR 下执行以下命令

```
ls * | awk -F '_PARTITION_' '{print $1}' | sort | uniq > /home/admin/version/TY/shell/combine/HX/table.conf
```

```
[root@dsjpt2 PAR]# pwd
/home/admin/version/TY/HX/json/Q/PAR
[root@dsjpt2 PAR]# ls * | awk -F '_PARTITION_' '{print $1}' | sort | uniq > /home/admin/version/TY/shell/combine/HX/table.conf
[root@dsjpt2 PAR]#
```

2、查看分区情况，可以在 odps 查看已同步的分区表的所有分区，都是以 P_开头

```
odps@ KF_JC_TY>show partitions T_TY_HX_SB_ZZS_LDSK;

rfq=P_1440000
rfq=P_1440100
rfq=P_1440102
rfq=P_1440103
rfq=P_1440104
rfq=P_1440105
rfq=P_1440106
rfq=P_1440107
rfq=P_1440111
rfq=P_1440112
rfq=P_1440115
rfq=P_1440116
rfq=P_1440171
rfq=P_1440181
rfq=P_1440182
rfq=P_1440183
rfq=P_1440184
```

如果不是，请更改脚本 shell/combine/ty_combine_partition.sh 如下图:rfq likt 'P_%', 将 P_改成实际情况，如电子退税则为 SYS_

```
#!/bin/bash
syscode=$1
bizdate=$2
#odpscmd="/home/sc_odps/odpscmd_public/bin/odpscmd"
odpscmd="/home/kf_odps/odpscmd_public/bin/odpscmd"
if [ -f ${syscode}/combine.sql ];then
    rm -f ${syscode}/combine.sql
fi
if [ -d ${syscode}/desc ];then
    rm -rf ${syscode}/desc
fi
mkdir ${syscode}/desc
for table in `cat ${syscode}/table.conf`;do
echo "table=${table}"
echo "alter table ${table} add if not exists partition(rfq='${bizdate}');" >> ${syscode}/combine.sql
${odpscmd} -e "desc ${table}" >${syscode}/desc/desc_${table}.log

selCols=`grep -E 'bigint|string|boolean|double|datetime|decimal|Partition' ${syscode}/desc/desc_${table}.log | awk '{print $2","}' | grep -vw 'Partition' | grep -vw 'rfq' | awk '{printf $0}'`
echo "insert overwrite table ${table} partition(rfq='${bizdate}') select ${selCols%?} from ${table} where rfq like 'P_%" >> ${syscode}/combine.sql
done
```

3、shell/combine 下执行脚本

./ty_combine_partition.sh HX 20171104

参数 20171104 为合并到哪个分区

```
[root@dsjpt2 combine]# pwd
/home/admin/version/TY/shell/combine
[root@dsjpt2 combine]# ./ty_combine_partition.sh HX 20171104
table=T_TY_HX_FP_YJJCYQK
OK
table=T_TY_HX_FP_YJJCYQK_MX
OK
table=T_TY_HX_SB_SBB
OK
table=T_TY_HX_SB_SBXX
OK
table=T_TY_HX_SB_YJSKXX
OK
table=T_TY_HX_SB_YJSKXXLSZ
OK
table=T_TY_HX_SB_YSBTJ
OK
table=T_TY_HX_SB_ZQJM
OK
table=T_TY_HX_SB_ZZS_LDSK
OK
table=T_TY_HX_SB_ZZS_LDSKLSZ
OK
table=T_TY_HX_SB_ZZS_YBNSR
OK
```

4、在 odps 客户端执行

```
./odpscmd -f /home/admin/version/TY/shell/combine/HX/combine.sql
```

```
[root@dsjpt2 bin]# ./odpscmd -f /home/admin/version/TY/shell/combine/HX/combine.sql
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
OK
```

5、删除原有分区

同理，请先确认 shell/combine/ty_drop_partition.sh 中 P_是否与实际的分区 P_开头一致


```
#!/bin/bash
odpscmd="/home/sc_odps/odpscmd_public/bin/odpscmd"
syscode=$1
if [ -f ${syscode}/dropPartitions.sql ];then
    rm -f ${syscode}/dropPartitions.sql
fi
if [ -d ${syscode}/par ];then
    rm -rf ${syscode}/par
fi
mkdir -p ${syscode}/par
for table in `cat ${syscode}/table.conf`;do
    echo "table=${table}"
    ${odpscmd} -e "show partitions ${table};" >>${syscode}/par/${table}.log
    for par in `cat ${syscode}/par/${table}.log|grep rfg=P_`;do
        partSpec=`echo ${par}|awk -F '=' '{print $1}'`
        echo "alter table ${table} drop if exists partition (${partSpec});">>${syscode}/dropPartitions.sql
    done
done
```

shell/combine 下执行脚本

./ty_drop_partition.sh HX

```
[root@dsjpt2 combine]# pwd
/home/admin/version/TY/shell/combine
[root@dsjpt2 combine]# ./ty_drop_partition.sh HX
table=T_TY_HX_FP_YJJCYQK

OK
table=T_TY_HX_FP_YJJCYQK_MX

OK
table=T_TY_HX_SB_SBB

OK
table=T_TY_HX_SB_SBXX

OK
table=T_TY_HX_SB_YJSKXX

OK
table=T_TY_HX_SB_YJSKXXLSZ

OK
table=T_TY_HX_SB_YSBTJ
```

在 odps 客户端执行

./odpscmd -f /home/admin/version/TY/shell/combine/HX/dropPartitions.sql

```
root@dsjpt2 bin]# ./odpscmd -f /home/admin/version/TY/shell/combine/HX/dropPartitions.sql

ID = 20171125085011548gailneb8
Log view:
http://logview.cn-guangzhou-gdgs-d01.odps.data.gdgs.sat.tax:9000/logview/?h=http://service.cn-guangzhou-gdgs-d01.odps.data.gdgs.sat.tax/api
6p=KF_JC_TY6i=20171125085011548gailneb8&token=VjIseIR3dzJlUwJY0UzdpbJVRU0I4R1Z4YU0BPSxpRFBTX09CTzpwNF8yMTYwODY4NTA4NzAsMTUxMjIwNDYxN
Sx7I1N0YXRlbWVudC16W3sIQW0aW9uIjpbIm9kCHM6UmVhZCJdLlJFZmZlY3Q101JBbGxvdyIsIlJlcz91cmNlIjpbImFjczpvZHBz01o6cHJvamVjdHMva2ZfamNfdHkvaW5zdgFu
Y2VzLzIwMTcxMTI1MDgIMDExNTQ4Z2Z6b65LYjgiXXI1dCJWZlZjZa9uIjoIMSJ9
Job Queuing.
OK

ID = 2017112508501759giblneb8
Log view:
http://logview.cn-guangzhou-gdgs-d01.odps.data.gdgs.sat.tax:9000/logview/?h=http://service.cn-guangzhou-gdgs-d01.odps.data.gdgs.sat.tax/api
6p=KF_JC_TY6i=2017112508501759giblneb8&token=K2tTR2QzTjVjWZDhrSkMzNkIxc2pxdJN6MzFBPSxpRFBTX09CTzpwNF8yMTYwODY4NTA4NzAsMTUxMjIwNDYxN
x7I1N0YXRlbWVudC16W3sIQW0aW9uIjpbIm9kCHM6UmVhZCJdLlJFZmZlY3Q101JBbGxvdyIsIlJlcz91cmNlIjpbImFjczpvZHBz01o6cHJvamVjdHMva2ZfamNfdHkvaW5zdgFu
Y2VzLzIwMTcxMTI1MDgIMDE3NTNlYmI5bWV0CJdVf0sIlZlcnNpbi24i0iXIn0=
Job Queuing.
OK

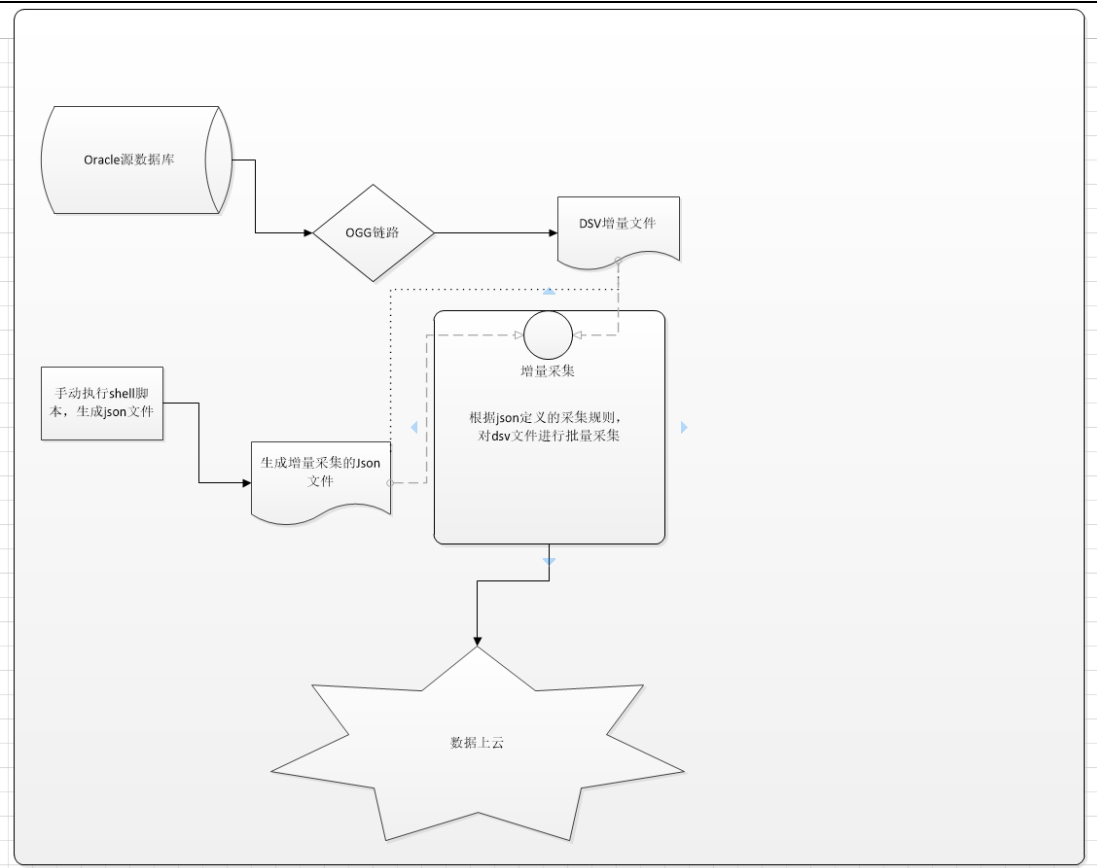
ID = 20171125085022363g8hbjeb8
Log view:
http://logview.cn-guangzhou-gdgs-d01.odps.data.gdgs.sat.tax:9000/logview/?h=http://service.cn-guangzhou-gdgs-d01.odps.data.gdgs.sat.tax/api
6p=KF_JC_TY6i=20171125085022363g8hbjeb8&token=0Eh3U1NMT6RaMDIPu9xz1dI7zm1KN3RdGfZPSxpRFBTX09CTzpwNF8yMTYwODY4NTA4NzAsMTUxMjIwNDYxN
ix7I1N0YXRlbWVudC16W3sIQW0aW9uIjpbIm9kCHM6UmVhZCJdLlJFZmZlY3Q101JBbGxvdyIsIlJlcz91cmNlIjpbImFjczpvZHBz01o6cHJvamVjdHMva2ZfamNfdHkvaW5zdgFu
Y2VzLzIwMTcxMTI1MDgIMDIyMzZzhoYmpLYjgiXXI1dCJWZlZjZa9uIjoIMSJ9
Job Queuing.
OK

ID = 20171125085027679gahbjeb8
Log view:
http://logview.cn-guangzhou-gdgs-d01.odps.data.gdgs.sat.tax:9000/logview/?h=http://service.cn-guangzhou-gdgs-d01.odps.data.gdgs.sat.tax/api
6p=KF_JC_TY6i=20171125085027679gahbjeb8&token=VElQ0FhTWjB2c2Y1am0wVudSTF20WkUzJpNPSxpRFBTX09CTzpwNF8yMTYwODY4NTA4NzAsMTUxMjIwNDYxN
yx7I1N0YXRlbWVudC16W3sIQW0aW9uIjpbIm9kCHM6UmVhZCJdLlJFZmZlY3Q101JBbGxvdyIsIlJlcz91cmNlIjpbImFjczpvZHBz01o6cHJvamVjdHMva2ZfamNfdHkvaW5zdgFu
Y2VzLzIwMTcxMTI1MDgIMDI3NjczZ2F0YmpLYjgiXXI1dCJWZlZjZa9uIjoIMSJ9
Job Queuing.
OK
```

2.3. Dax 增量初始化实施

2.3.1. 总体流程及结构

2.3.1.1. 流程图



2.3.1.2. 具体步骤

1. 在源端以及目标端配置 ogg 投递和接收进程(具体请参考 ogg 实施文档)。
2. 配置 ty_datasource.conf 该文件主要是配置来源数据库信息、json 生成规则 (具体参考核心征管)。

目录为 TY/HX/conf, 根据现场实际情况配置以下信息, 主要包括 odps、oracle、ftp 服务器相关信息

```

#####
##
#ODPS 配置          --下面部分都是 ODPS 配置的内容
#####
##
odpsServer="http://service.odps.aliyun.com/api"          #ODPS 服务器地址
tunnelServer=""          #ODPS tunnel 地址(未使用)
odpsaccessId="***** "          #ODPS 的 accessid
odpsaccessKey="***** "          #ODPS 的 accesskey
project="SC_JC_TY"#项目名称
errorLimit=10#允许错误记录数(未使用)
dataxLog="/home/datax/log"#datax 日志目录(未使用)

```

```
#####
##
#源库配置
#####
##
#数据读取类型: oraclereader、mysqlreader
reader="oraclereader"
#reader="mysqlreader"
jdbc="xx.xx.xx.xx:1521/xxxx" #连接源库 jdbc 地址
user="dsjsy" #源库用户名
pass="*****" #源库密码
odpsJdbc="jdbc:oracle:thin:@${jdbc}" #连接 odps 的 jdbc
#####
##
#文件服务器地址，用于存储增量 dsv 文件
#####
##
ftpIp="xx.xx.xx.x" ##ftp 服务器地址
ftpPort="22" ##ftp 服务器地址
ftpUser="root" ##ftp 服务器用户名
ftpPasswd="*****" ##ftp 服务器密码
ftpBasePath="/home/DSV/HX"
#####
##
#其他配置
#####
##
##是否开启版本控制
checkVersion=false ##开启版本控制(未使用)
##ogg 增量文件后缀
oggSubfix="dsv"
##ogg 增量文件分隔符
spiltFlag="@@"
##oracle_home 配置
TY_ORACLE_HOME="/oradata/app/oracle/product/11.2.0/db_1"
##oracle_lib 配置
TY_LD_LIBRARY_PATH="${TY_ORACLE_HOME}/lib"
#####
##
```

在 ty_datasource.conf 中，需要配置的信息有，odps 服务端连接信息，来源库连接信息，文档服务器 ftp 连接信息，oracle 环境变量设置等，目前在 xx.xx.xx.xx 上可通过修改来源库连接信息复用同个 ty_datasource 配置文件。

- 配置 `ty_createJson_zl.conf` 该文件主要是配置需要增量采集的数据表(具体参考核心征管)。

目录为 `TY/HX/conf`, 此文件需要根据数据字典手动配置, 一张表对应一行数据, 格式为: 域名|表名|odps 表名|主键|是否为分区表|表记录数

```
#####域名|表名|odps 表名|主键|是否为分区表|表记录数|
HX_CS_QG|CS_DJ_SXXDGDQGLB|T_TY_HX_CS_DJ_SXXDGDQGLB|SSXD_DM|NO|159|
HX_CS_QG|CS_DJ_SWSZZJLXDZB|T_TY_HX_CS_DJ_SWSZZJLXDZB|SWZJZL_DM|NO|46|
```

```
1 用户域名|数据表名|odps表名|主键|是否为分区表|表数据量|where条件
2 HX_CS_QG|CS_DJ_SXXDGDQGLB|T_TY_HX_CS_DJ_SXXDGDQGLB|SSXD_DM|NO|159|
3 HX_CS_QG|CS_DJ_SWSZZJLXDZB|T_TY_HX_CS_DJ_SWSZZJLXDZB|SWZJZL_DM|NO|46|
4 HX_CS_QG|CS_DJ_YGZSDZSPMYZSPMDZB|T_TY_HX_CS_DJ_YGZSDZSPMYZSPMDZB|UUID|NO|84|
5 HX_CS_QG|CS_DJ_ZDSYJKJCDZB|T_TY_HX_CS_DJ_ZDSYJKJCDZB|JGCJ_DM|NO|5|
6 HX_CS_QG|CS_DJ_ZJZLDJZCLXDZB|T_TY_HX_CS_DJ_ZJZLDJZCLXDZB|SFZJZL_DM|NO|30|
7 HX_CS_QG|CS_FP_FPDKKXZSPM|T_TY_HX_CS_FP_FPDKKXZSPM|ZSPM_DM|NO|10|
8 HX_CS_QG|CS_FP_HWYSZPBKXZSPM|T_TY_HX_CS_FP_HWYSZPBKXZSPM|ZSXM_DM|NO|167|
9 HX_CS_QG|CS_FP_YGZJYLBZSPMDZ|T_TY_HX_CS_FP_YGZJYLBZSPMDZ|ZSPM_DM|NO|187|
10 HX_CS_QG|CS_GY_XJDMZDZB|T_TY_HX_CS_GY_XJDMZDZB|UUID|NO|189|
11 HX_CS_QG|CS_JC_SSWFXWCLCFYJDZB|T_TY_HX_CS_JC_SSWFXWCLCFYJDZB|UUID|NO|3412|
12 HX_CS_QG|CS_JC_WFSDYJCFGX|T_TY_HX_CS_JC_WFSDYJCFGX|WFWZXW_DM|NO|94|
13 HX_CS_QG|CS_PZ_PZFLB|T_TY_HX_CS_PZ_PZFLB|PZFL_DM|NO|15|
14 HX_CS_QG|CS_PZ_PZZLFLGLB|T_TY_HX_CS_PZ_PZZLFLGLB|PZZLFLGLUID|NO|58|
15 HX_CS_QG|CS_RD_SXXSDSLXSLDZB|T_TY_HX_CS_RD_SXXSDSLXSLDZB|SXXSDSLX_DM|NO|14|
16 HX_CS_QG|CS_RD_XZQHXYXZDZB|T_TY_HX_CS_RD_XZQHXYXZDZB|XZQHSZ_DM|NO|56|
17 HX_CS_QG|CS_RD_ZZSJYBFZSHWZSLDZB|T_TY_HX_CS_RD_ZZSJYBFZSHWZSLDZB|JYBFZSZSHWLX_DM|NO|48|
18 HX_CS_QG|CS_SB_HSQJYSKMDZB|T_TY_HX_CS_SB_HSQJYSKMDZB|XH|NO|15|
19 HX_CS_QG|CS_SB_HYHDLRLDZB|T_TY_HX_CS_SB_HYHDLRLDZB|HY_DM|NO|4|
20 HX_CS_QG|CS_SB_KQSYFFL|T_TY_HX_CS_SB_KQSYFFL|ZSPM_DM|NO|20|
21 HX_CS_QG|CS_SB_MSSR|T_TY_HX_CS_SB_MSSR|UUID|NO|7|
22 HX_CS_QG|CS_SB_QSQSZYYZSPMDZB|T_TY_HX_CS_SB_QSQSZYYZSPMDZB|UUID|NO|115|
23 HX_CS_QG|CS_SB_QYSDS_HYDZB|T_TY_HX_CS_SB_QYSDS_HYDZB|UUID|NO|100|
24 HX_CS_QG|CS_SB_SBYHJMZSBMRX|T_TY_HX_CS_SB_SBYHJMZSBMRX|ZSBYWL|NO|24|
25 HX_CS_QG|CS_SB_SDSSWSXDZB|T_TY_HX_CS_SB_SDSSWSXDZB|HC|NO|206|
26 HX_CS_QG|CS_SB_SSYHXMJMXZDZB|T_TY_HX_CS_SB_SSYHXMJMXZDZB|SSYHXM|NO|19|
27 HX_CS_QG|CS_SB_TSNRTSGZDZB|T_TY_HX_CS_SB_TSNRTSGZDZB|TSNRLX_DM|NO|3|
28 HX_CS_QG|CS_SB_WSSBPZZLDZB|T_TY_HX_CS_SB_WSSBPZZLDZB|YZPZZL_DM|NO|79|
29 HX_CS_QG|CS_SB_YZPZZLFDZXX|T_TY_HX_CS_SB_YZPZZLFDZXX|YZPZZL_DM|NO|83|
30 HX_CS_QG|CS_YH_CLGZSMSTJLSWSXDZB|T_TY_HX_CS_YH_CLGZSMSTJLSWSXDZB|SLWSX_DM|NO|18|
31 HX_CS_QG|CS_YH_FJMXDSL|T_TY_HX_CS_YH_FJMXDSL|UUID|NO|987|
32 HX_CS_QG|CS_YH_MSLB|T_TY_HX_CS_YH_MSLB|MSLB_DM|NO|9|
33 HX_CS_QG|CS_YH_MSMX|T_TY_HX_CS_YH_MSMX|MSMX_DM|NO|242|
34 HX_CS_QG|CS_YH_SWSXYZCSLBDZB|T_TY_HX_CS_YH_SWSXYZCSLBDZB|UUID|NO|39|
35 HX_CS_QG|CS_YH_SWSXZSFSPDZB|T_TY_HX_CS_YH_SWSXZSFSPDZB|SWSX_DM|NO|87|
36 HX_CS_QG|CS_YH_SWSXZSPMHC|T_TY_HX_CS_YH_SWSXZSPMHC|ZSPM_DM|NO|6|
37 HX_CS_QG|CS_YH_TDJMSWSX|T_TY_HX_CS_YH_TDJMSWSX|SWSX_DM|NO|19|
38 HX_CS_QG|CS_YH_ZCSSZJFLNXDZB|T_TY_HX_CS_YH_ZCSSZJFLNXDZB|ZCSSZJFL_DM|NO|5|
```

在 `ty_createJson_al.conf` 中, 需要配置准备增量采集的数据表的元信息, 按顺序将用户域名|数据库表名|odps 表名|主键|是否为分区表|表数据量|where 条件, 其中 where 条件置空, 因为目前增量采集的方式是通过采集 ogg 生产的 dsv 文件。

- 运行 shell 脚本 `ty_dsv_to_datax_zl.sh` HX 产出 json 文件位于 `admin/version/TY/HX/json/DSV`。
- 拷贝第四步骤的执行脚本和所有 json 到两台 gateway 机器上, 参考命令:

```
scp -r ty_commit_datax_zl.sh root@xxx.xx.xx.xx:/home/admin/vesion/TY/shell
```

```
scp -r ty_commit_datax_zl.sh root@xxx.xx.xx.xx:/home/admin/vesion/TY/shell
```

```
scp -r DSV/* root@xxx.xx.xx.xx:/home/admin/vesion/TY/HX/json/DSV
```

```
scp -r DSV/* root@xxx.xx.xx.xx:/home/admin/vesion/TY/HX/json/DSV
```

- 配置 Excel 增量任务, 拷贝该 Excel 到 `xxx.xx.xx.xx` 远程机上, 运行 python 程序生成阿里云大数据平台的增量调度任务。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
22	HY	T.TV.HY.DJ.SKSJKDK.KSI.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
23	HY	T.TV.HY.DJ.SKSJKDK.LS.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
24	HY	T.TV.HY.DJ.SKSJKDK.LS.ZB.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
25	HY	T.TV.HY.DJ.SKSJKDK.NSDAOXTS.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
26	HY	T.TV.HY.DJ.SKSJKDK.SKPOL.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
27	HY	T.TV.HY.DJ.SKSJKDK.SKSXOTS.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
28	HY	T.TV.HY.DJ.SKSJKDK.SZSM.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
29	HY	T.TV.HY.DJ.SKSJKDK.YHMH.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
30	HY	T.TV.HY.DJ.SKSJKDK.YHSWOK.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
31	HY	T.TV.HY.DJ.SKSJKDK.YHZCBG.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
32	HY	T.TV.HY.DJ.SKSJKDK.YHZX.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
33	HY	T.TV.HY.DJ.SKSJKDK.YHZX.ZB.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
34	HY	T.TV.HY.DJ.SKSJKDK.YHGB.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
35	HY	T.TV.HY.DJ.SKSJKDK.ZCBG.CSP.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
36	HY	T.TV.HY.DJ.SKSJKDK.ZCBG.FP.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
37	HY	T.TV.HY.DJ.SKSJKDK.ZCBG.JKSJ.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
38	HY	T.TV.HY.DJ.SKSJKDK.ZCBG.SKP.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
39	HY	T.TV.HY.DJ.SKSJKDK.ZCBG.SWOK.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
40	HY	T.TV.HY.DJ.SKSJKDK.ZCBG.SZSM.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
41	HY	T.TV.HY.DJ.SKSJKDK.ZCBG.ZYH.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
42	HY	T.TV.HY.DJ.SKSJKDK.ZCBGOM.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
43	HY	T.TV.HY.DJ.SKSJKDK.ZYHOK.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
44	HY	T.TV.HY.DM.DNSR.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
45	HY	T.TV.HY.DM.GV.DIZCLX.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
46	HY	T.TV.HY.DM.GV.FP.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
47	HY	T.TV.HY.DM.GV.FPZL.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
48	HY	T.TV.HY.DM.GV.HY.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
49	HY	T.TV.HY.DM.GV.JDXZ.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
50	HY	T.TV.HY.DM.GV.NSQS.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
51	HY	T.TV.HY.DM.GV.SBSPS.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
52	HY	T.TV.HY.DM.GV.SIBSX.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
53	HY	T.TV.HY.DM.GV.SIBSX.R.Z		knjd_etl_hy_mr	/内部数据/货运发货	hy_commit_datax_zi.sh									
	V1	V2	SHELL_R_Z	SHELL_Q											

从截图可以看出，Excel 配置文件有 4 个 sheet 页面，分别是 V1、V2、SHELL_R_Z、SHELL_Q,其中，V1 和 V2 页面是用于创建虚拟节点，配置信息可复用，SHELL_R_Z 是用于创建增量任务，具体的配置项有

配置项	配置信息	备注
SYSCODE	业务系统代码简写	
TASK_NAME	任务名称	
TASK_COMMENT	任务	
PRE_TASK_NAME	父节点	
FOLDER_NAME	存放目录	
SCRIPT_NAME	执行脚本	

7. 执行 Python 程序生成增量调度任务

由于 python 程序是由阿里提供，importBatch.py 和 trunBatch.py，功能分别是根据 excel 配置创建增量任务，根据 excel 配置批量删除增量任务。以下对 python 进行解释说明：

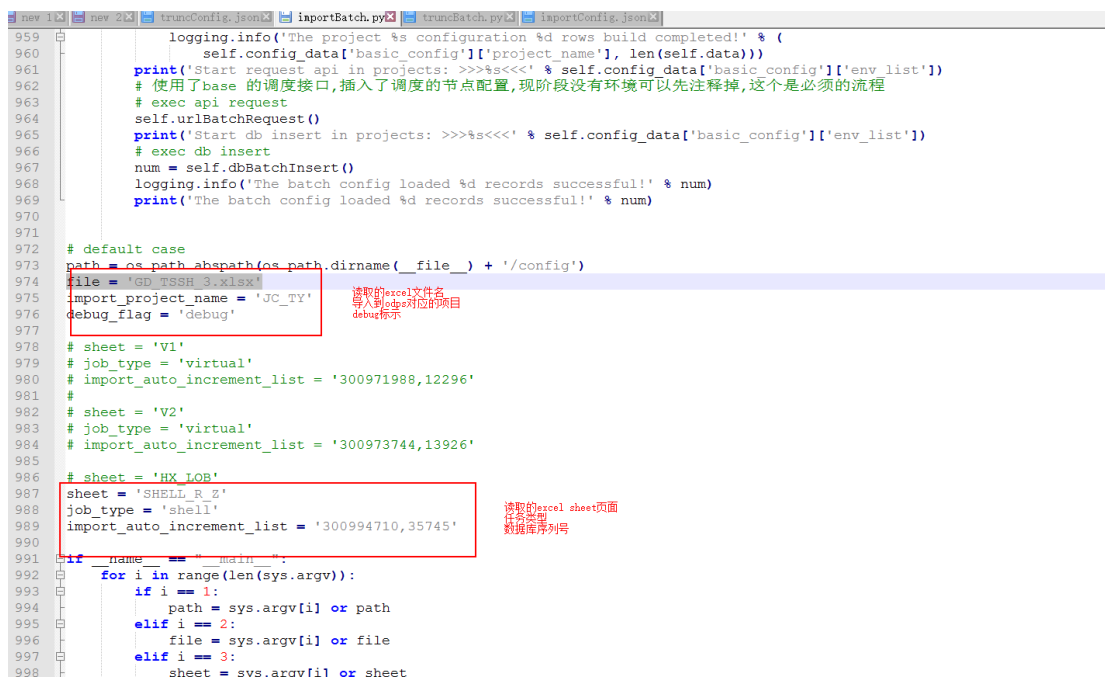
- 对于 importBatch.py 脚本，需要对以下红框截图内容进行修改，参考如下：
对于需要修改的 import_auto_incremenet_list 变量，需要先登录跳板机，执行查询语句，查询出数据库序列号步骤参考如下：

步骤	备注
登录跳板机	命令：ssh xxx.xx.xx.xx 密码：*****
登录 mysql 数据库	mysql -hdpbize.mysql.data.gdgs.sat.tax -udpbize -

	p2HEswtesaxoiwey3
查询 1	select max(id) from (select max(id) id from dpbizide.ide_base_object union all select max(id) id from dpbizide.ide_flow union all select max(child_id) id from dpbizide.ide_flow_node_relation)t;
查询 2	select max(id) from dpbizide.ide_node_version;

➤ 对于 importBatch.py 脚本，需要对以下红框截图内容进行修改，参考如下：

对于红框中 6 个参数的含义为：读取解析的 excel 文件、批量导入 odps 对应的项目名、debug 模式、读取的 excel 文件 sheet 页面、任务类型、数据库序列号



```

959 logging.info('The project %s configuration %d rows build completed!' % (
960     self.config_data['basic_config']['project_name'], len(self.data)))
961 print('Start request api in projects: >>>%s<<<' % self.config_data['basic_config']['env_list'])
962 # 使用了base 的调度接口,插入了调度的节点配置,现阶段没有环境可以先注释掉,这个是必须的流程
963 # exec api request
964 self.urlBatchRequest()
965 print('Start db insert in projects: >>>%s<<<' % self.config_data['basic_config']['env_list'])
966 # exec db insert
967 num = self.dbBatchInsert()
968 logging.info('The batch config loaded %d records successful!' % num)
969 print('The batch config loaded %d records successful!' % num)
970
971
972 # default case
973 path = os.path.abspath(os.path.dirname(__file__) + '/config')
974 file = 'GD_TSSH_3.xlsx'
975 import_project_name = 'JC_TY'
976 debug_flag = 'debug'
977
978 # sheet = 'V1'
979 # job_type = 'virtual'
980 # import_auto_increment_list = '300971988,12296'
981 #
982 # sheet = 'V2'
983 # job_type = 'virtual'
984 # import_auto_increment_list = '300973744,13926'
985 #
986 # sheet = 'HX_LOB'
987 sheet = 'SHELL_R_Z'
988 job_type = 'shell'
989 import_auto_increment_list = '300994710,35745'
990
991 if __name__ == '__main__':
992     for i in range(len(sys.argv)):
993         if i == 1:
994             path = sys.argv[i] or path
995         elif i == 2:
996             file = sys.argv[i] or file
997         elif i == 3:
998             sheet = sys.argv[i] or sheet
    
```

读取的 excel 文件名
导入到 odps 对应的项目
debug 标识

读取的 excel sheet 页面
任务类型
数据库序列号

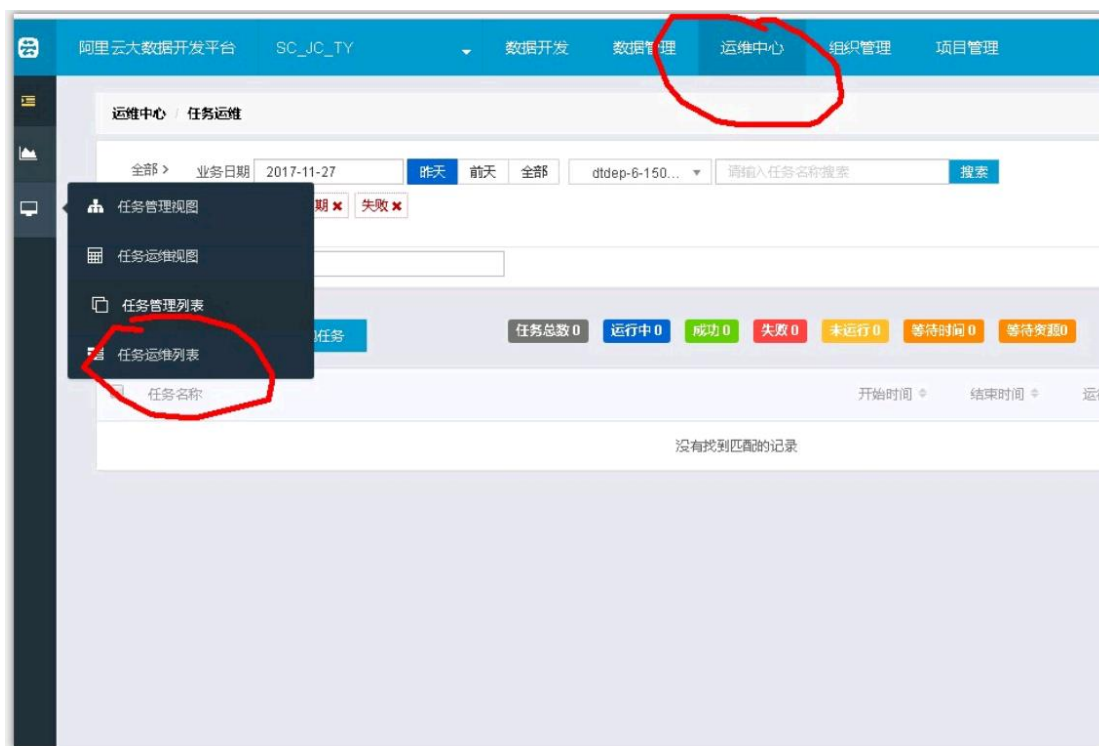
➤ 对于 truncBatch.py 脚本，需要对以下红框截图内容进行修改，参考如下：

对于红框中 4 个参数的含义为：读取解析的 excel 文件、读取的 excel 文件 sheet 页面、任务类型、批量删除的 odps 项目名


```
387         oneData[self.config_data['field_mapping']['task_name']].lower())
388         deleteNodeUrlList.append((truncUrl, 'DELETE', oneRow, project_name))
389
390     try:
391         pool = ThreadPool(self.config_data['web']['multi_thread'])
392         pool.map(self.truncBatchParamsTransform, deleteNodeUrlList)
393         pool.close()
394         pool.join()
395         logging.info('Delete config {rows} rows successful!'.format(rows=len(deleteNodeUrlList)))
396     except:
397         logging.error('Delete error when creating pool!')
398         raise requests.exceptions.RequestException('Request base api error!')
399
400
401 # default case
402 path = os.path.abspath(os.path.dirname(__file__) + '/config')
403 file = 'GD_TSSH_3.xlsx'
404 sheet = 'SHELL_R_Z'
405 # sheet = 'V2'
406 job_type = 'shell'
407
408 trunc_project_name = 'JC_TY'
409 debug_flag = ''
410
411 if __name__ == "__main__":
412     for i in range(len(sys.argv)):
413         if i == 1:
414             path = sys.argv[i] or path
415         elif i == 2:
416             file = sys.argv[i] or file
417         elif i == 3:
418             sheet = sys.argv[i] or sheet
419         elif i == 4:
420             job_type = sys.argv[i] or job_type
421         elif i == 5:
422             trunc_project_name = sys.argv[i] or trunc_project_name
423         elif i == 6:
424             debug_flag = sys.argv[i]
425     if path[-1] != '\\':
426         path += '\\'
```

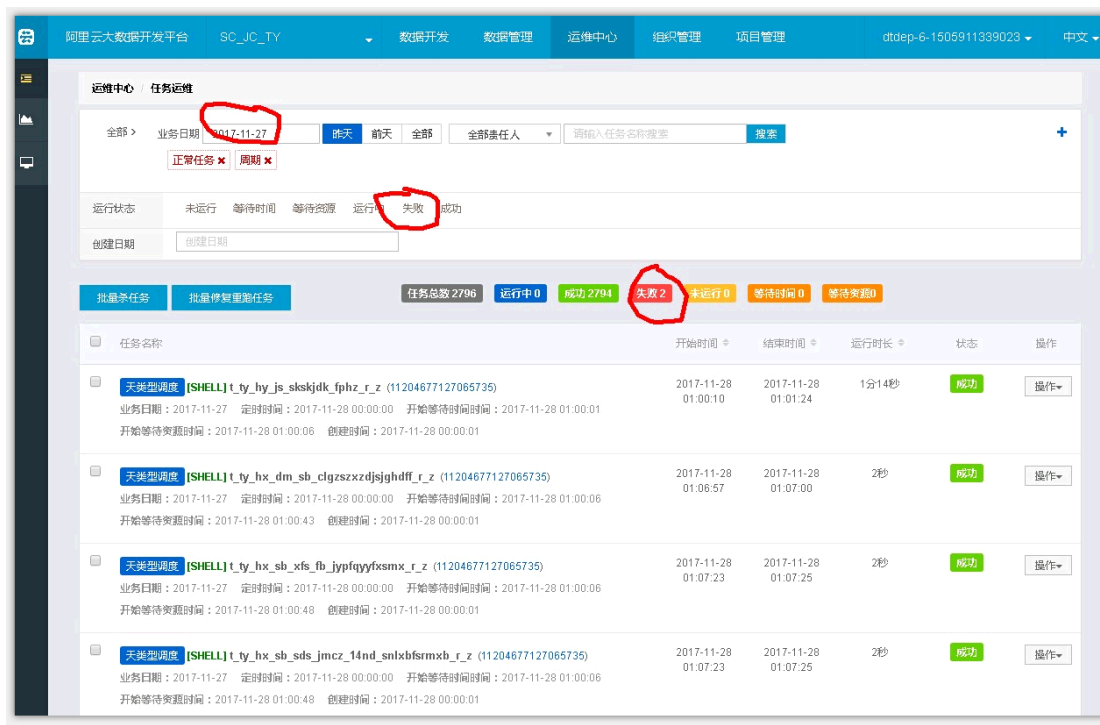
读取的excel文件
读取的excel sheet页面
任务类型
需要删除的项目名

8. 在阿里云大数据开发平台运维增量采集任务，查看是否成功生成了增量任务
- 登录阿里云大数据管理平台，如下图所示进入运维中心查看增量任务执行情况，操作步骤是：点击运维中心选项卡，进入任务运维列表。

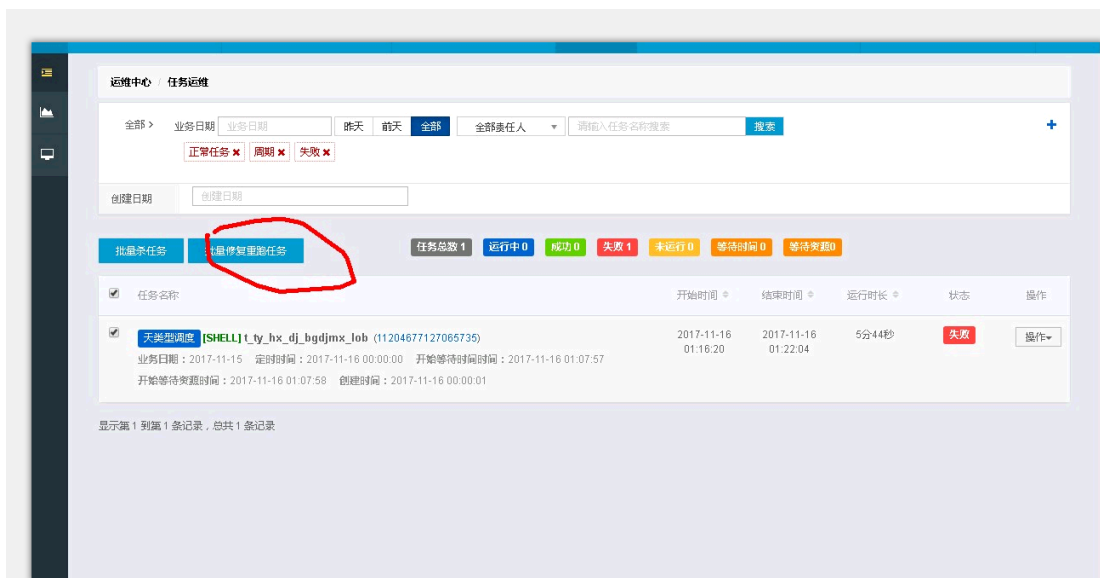


- 根据业务日期筛选出当天执行的任务，或者根据任务名称匹配出任务列表，运行状态选中失败一项，然后点击搜索出当天失败的任务列表，分析任务失败

的具体原因，一般情况下，需要远程连接 getway 机器查看任务的失败原因。



➤ 批量选中失败任务，直接选中批量修复任务，对任务进行重跑。

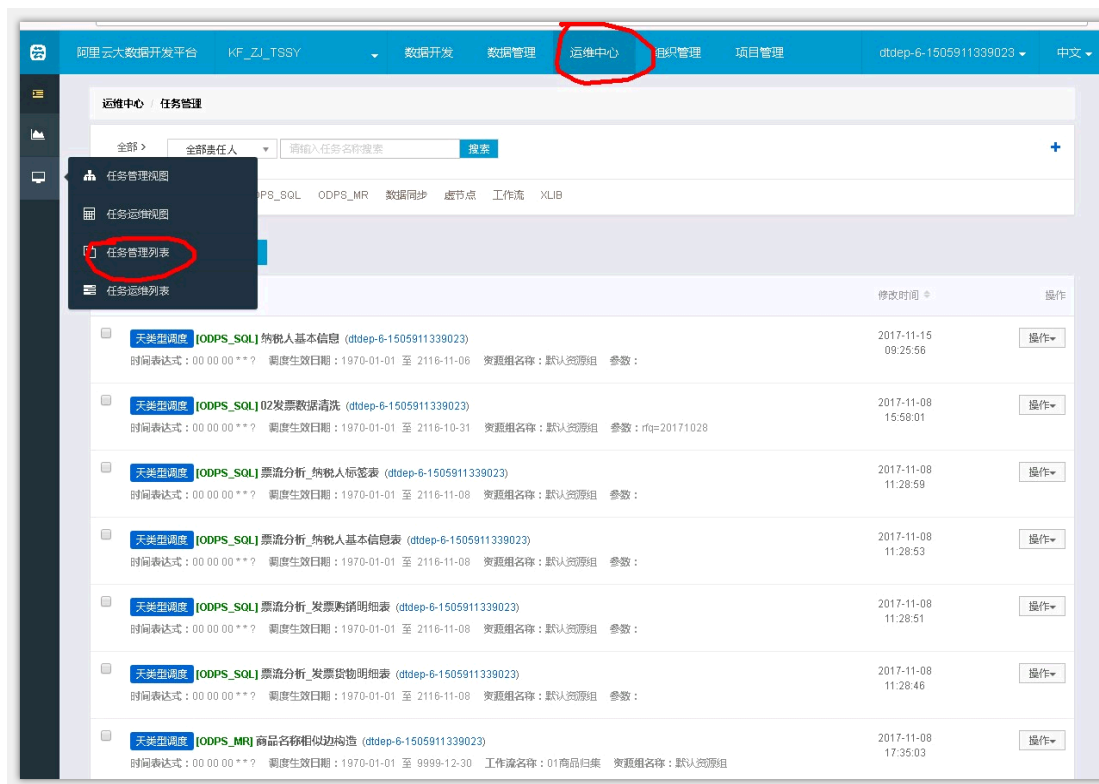


2.3.1.3. 补数据

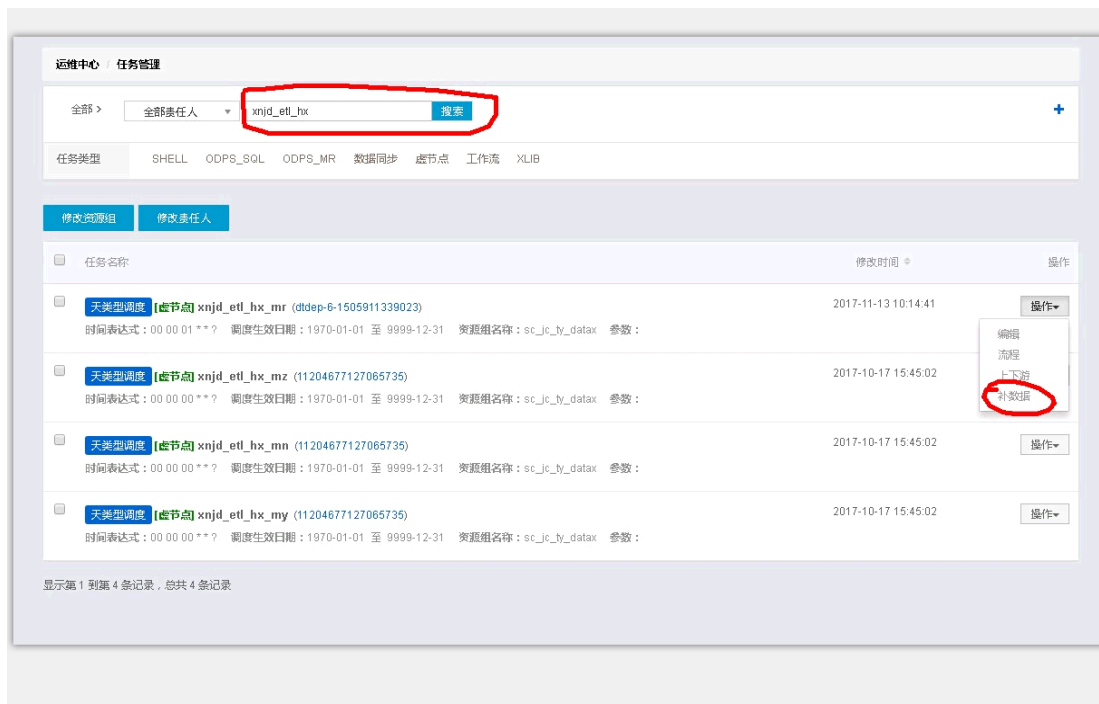
数据上云过程中，对于数据丢失、数据错误、重新采集某个业务日期的增量数据，可以采用平台提供的补数据功能，具体步骤是，打开运维中心，选择任务管理列表，通过关键字和业务日期搜索查询出需要补数据的任务，详细如下图所示

示:

- 打开运维中心，选择任务管理列表，通过搜索关键字查询出需要补数据的任务

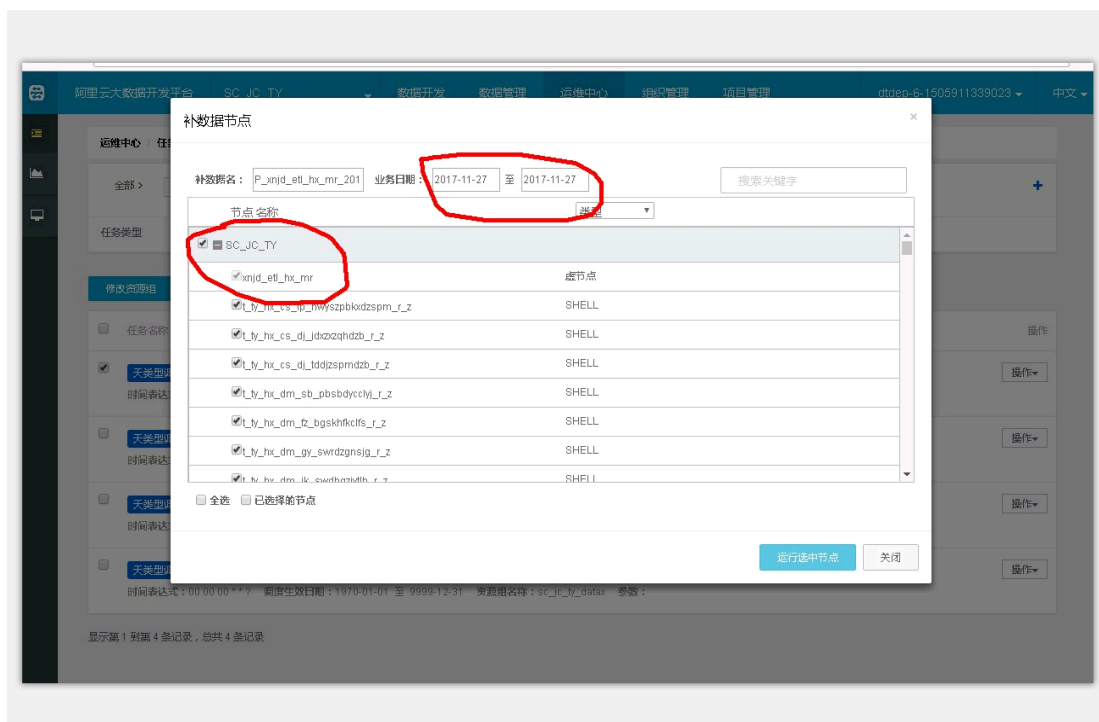


- 打开操作下拉框，选中补数据



- 选中需要补数据的业务日期，勾选需要补数据的任务列表，然后点击运行选

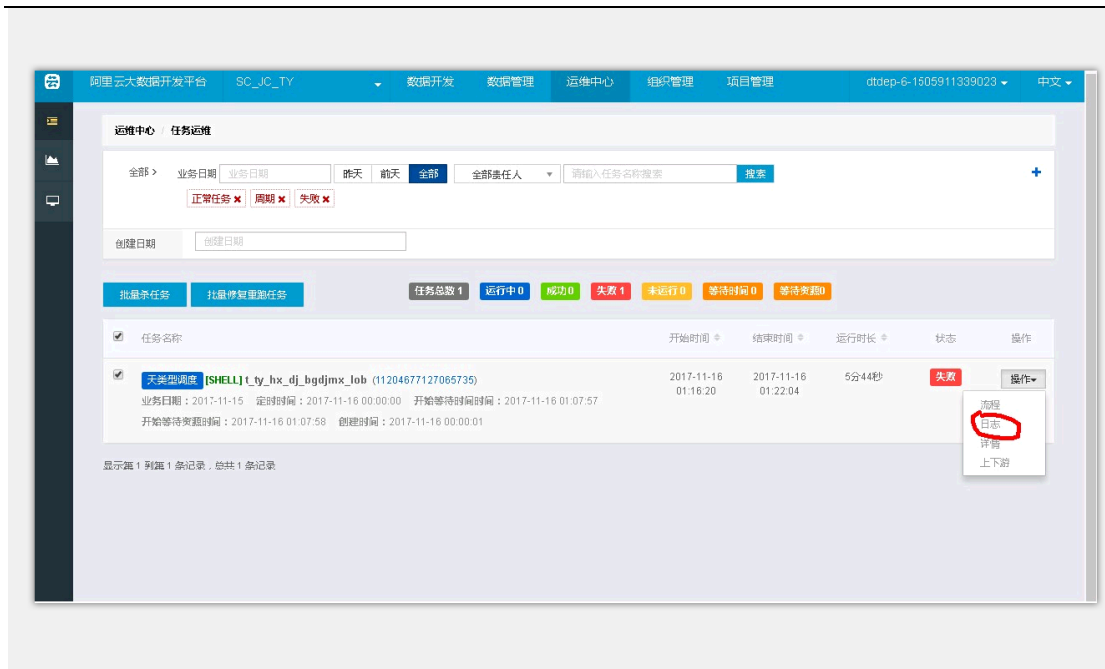
中节点按钮开始补数据



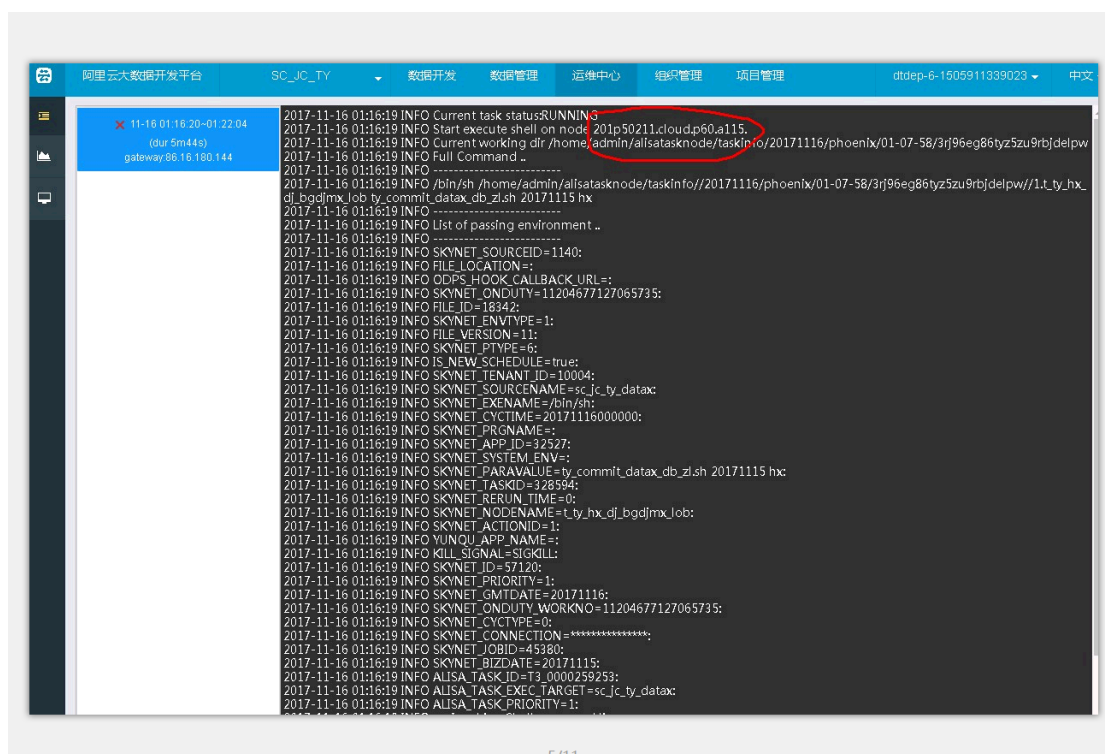
2.3.1.4. 分析失败任务

对于增量同步失败的任务，由于 **dsv** 源数据文件存在乱码或者源端增减字段的原因，直接重跑可能还是会失败，所以，需要通过查看任务的执行日志分析出具体的失败原因，大概的步骤流程是，通过阿里云大数据开发平台定位出增量任务的执行节点，然后远程登录该节点，查看失败日志，分析出失败原因，如下图案例所示：

- 在运维中心的任务运维列表中搜索出当天执行失败的任务，点开操作下拉框，点击日志：



- 在新开的页面中，注意下图红框位置的日志，此处说明了任务的执行节点位于 201p50211.cloud.p60.a115 机器，通过 xshell 远程登录该机器



- 如下图所示，日志存放目录位于 /home/admin/version/TY/shell/log/commit/，根据任务执行日期，进入相应的目录

```
#ll
total 132
drwxrwxr-x 5 admin admin 4096 Oct 27 19:52 20171027 ✓
drwxrwxr-x 6 admin admin 4096 Oct 28 09:22 20171028
drwxrwxr-x 6 admin admin 4096 Oct 29 01:00 20171029
drwxrwxr-x 7 admin admin 4096 Oct 30 14:23 20171030
drwxrwxr-x 6 admin admin 4096 Oct 31 01:00 20171031
drwxrwxr-x 6 admin admin 4096 Nov 1 01:00 20171101
drwxrwxr-x 6 admin admin 4096 Nov 2 01:00 20171102
drwxrwxr-x 8 admin admin 4096 Nov 3 17:49 20171103
drwxrwxr-x 7 admin admin 4096 Nov 4 09:33 20171104
drwxrwxr-x 7 admin admin 4096 Nov 5 01:00 20171105
drwxrwxr-x 7 admin admin 4096 Nov 6 01:00 20171106
drwxrwxr-x 8 admin admin 4096 Nov 7 16:41 20171107
drwxrwxr-x 7 admin admin 4096 Nov 8 01:00 20171108
drwxrwxr-x 7 admin admin 4096 Nov 9 01:00 20171109
drwxrwxr-x 7 admin admin 4096 Nov 10 01:00 20171110
drwxrwxr-x 8 admin admin 4096 Nov 11 17:10 20171111
drwxrwxr-x 7 admin admin 4096 Nov 12 01:00 20171112
drwxrwxr-x 7 admin admin 4096 Nov 13 01:00 20171113
drwxrwxr-x 7 admin admin 4096 Nov 14 01:02 20171114
drwxrwxr-x 7 admin admin 4096 Nov 15 01:00 20171115
drwxrwxr-x 7 admin admin 4096 Nov 16 01:04 20171116
drwxrwxr-x 7 admin admin 4096 Nov 17 01:00 20171117
drwxrwxr-x 7 admin admin 4096 Nov 18 01:00 20171118
drwxrwxr-x 7 admin admin 4096 Nov 19 01:00 20171119
drwxrwxr-x 7 admin admin 4096 Nov 20 01:00 20171120
drwxrwxr-x 7 admin admin 4096 Nov 21 01:00 20171121
drwxrwxr-x 7 admin admin 4096 Nov 22 01:00 20171122
drwxrwxr-x 7 admin admin 4096 Nov 23 01:04 20171123
drwxrwxr-x 7 admin admin 4096 Nov 24 01:00 20171124
drwxrwxr-x 7 admin admin 4096 Nov 25 01:00 20171125
drwxrwxr-x 7 admin admin 4096 Nov 26 01:00 20171126
drwxrwxr-x 7 admin admin 4096 Nov 27 01:00 20171127
drwxrwxr-x 7 admin admin 4096 Nov 28 01:00 20171128

[root@201p43112.cloud.p44.a115 /home/admin/version/TY/shell/log/commit]
# cd /home/admin/version/TY/shell/log/commit
[root@201p43112.cloud.p44.a115 /home/admin/version/TY/shell/log/commit]
```

- 如下图，编辑具体的失败日志，可清晰看到该任务的失败原因


```
total 667456
-rw-rw-r-- 1 admin admin 12553006 Oct 27 19:53 T_TY_DZ_DZDZ_DA_XGMNSR_HX_SM_DSV_R_Z_195255.log
-rw-rw-r-- 1 admin admin 12553476 Oct 27 19:53 T_TY_DZ_DZDZ_DA_XGMNSR_HX_SM_DSV_R_Z_195309.log
-rw-rw-r-- 1 admin admin 12553628 Oct 27 19:57 T_TY_DZ_DZDZ_DA_XGMNSR_HX_SM_DSV_R_Z_195706.log
-rw-rw-r-- 1 admin admin 12553234 Oct 27 19:57 T_TY_DZ_DZDZ_DA_XGMNSR_HX_SM_DSV_R_Z_195720.log
-rw-rw-r-- 1 admin admin 22633 Oct 27 19:53 T_TY_DZ_DZDZ_DA_XGMNSR_ZC_DSV_R_Z_195255.log
-rw-rw-r-- 1 admin admin 26061 Oct 27 19:53 T_TY_DZ_DZDZ_DA_XGMNSR_ZC_DSV_R_Z_195321.log
-rw-rw-r-- 1 admin admin 17550996 Oct 27 19:53 T_TY_DZ_DZDZ_DA_YBNSR_ZC_DSV_R_Z_195255.log
-rw-rw-r-- 1 admin admin 17543268 Oct 27 19:53 T_TY_DZ_DZDZ_DA_YBNSR_ZC_DSV_R_Z_195317.log
-rw-rw-r-- 1 admin admin 17543014 Oct 27 19:57 T_TY_DZ_DZDZ_DA_YBNSR_ZC_DSV_R_Z_195706.log
-rw-rw-r-- 1 admin admin 17543014 Oct 27 19:57 T_TY_DZ_DZDZ_DA_YBNSR_ZC_DSV_R_Z_195717.log
-rw-rw-r-- 1 admin admin 65534 Oct 27 19:53 T_TY_DZ_DZDZ_DHX_FPXX_DSV_R_Z_195255.log
-rw-rw-r-- 1 admin admin 65272 Oct 27 19:54 T_TY_DZ_DZDZ_DHX_FPXX_DSV_R_Z_195339.log
-rw-rw-r-- 1 admin admin 65534 Oct 27 19:57 T_TY_DZ_DZDZ_DHX_FPXX_DSV_R_Z_195706.log
-rw-rw-r-- 1 admin admin 65270 Oct 27 19:58 T_TY_DZ_DZDZ_DHX_FPXX_DSV_R_Z_195745.log
-rw-rw-r-- 1 admin admin 45702487 Oct 27 19:53 T_TY_DZ_DZDZ_FPXX_JDCFP_DSV_R_Z_195255.log
-rw-rw-r-- 1 admin admin 45702486 Oct 27 19:53 T_TY_DZ_DZDZ_FPXX_JDCFP_DSV_R_Z_195326.log
-rw-rw-r-- 1 admin admin 29850 Oct 27 19:53 T_TY_DZ_DZDZ_FPXX_PTFP_DSV_R_Z_195255.log
-rw-rw-r-- 1 admin admin 28677 Oct 27 19:53 T_TY_DZ_DZDZ_FPXX_PTFP_DSV_R_Z_195324.log
-rw-rw-r-- 1 admin admin 89108592 Oct 27 19:53 T_TY_DZ_DZDZ_FPXX_ZF_DSV_R_Z_195255.log
-rw-rw-r-- 1 admin admin 89114304 Oct 27 19:53 T_TY_DZ_DZDZ_FPXX_ZF_DSV_R_Z_195323.log
-rw-rw-r-- 1 admin admin 8911595 Oct 27 19:57 T_TY_DZ_DZDZ_FPXX_ZF_DSV_R_Z_195706.log
-rw-rw-r-- 1 admin admin 58895133 Oct 27 19:57 T_TY_DZ_DZDZ_FPXX_ZF_DSV_R_Z_195723.log
-rw-rw-r-- 1 admin admin 28068 Oct 27 19:53 T_TY_DZ_DZDZ_FPXX_ZZSFP_DSV_R_Z_195255.log
-rw-rw-r-- 1 admin admin 25189 Oct 27 19:53 T_TY_DZ_DZDZ_FPXX_ZZSFP_DSV_R_Z_195316.log
-rw-rw-r-- 1 admin admin 28067 Oct 27 19:57 T_TY_DZ_DZDZ_FPXX_ZZSFP_DSV_R_Z_195706.log
-rw-rw-r-- 1 admin admin 27354 Oct 27 19:57 T_TY_DZ_DZDZ_FPXX_ZZSFP_DSV_R_Z_195723.log
-rw-rw-r-- 1 admin admin 21406 Oct 27 19:57 T_TY_DZ_DZDZ_HWXX_DZFP_DSV_R_Z_195706.log
-rw-rw-r-- 1 admin admin 21899 Oct 27 19:57 T_TY_DZ_DZDZ_HWXX_DZFP_DSV_R_Z_195727.log
-rw-rw-r-- 1 admin admin 24032570 Oct 27 19:53 T_TY_DZ_DZDZ_HZFP_TZD_ZB_DSV_R_Z_195255.log
-rw-rw-r-- 1 admin admin 24032570 Oct 27 19:53 T_TY_DZ_DZDZ_HZFP_TZD_ZB_DSV_R_Z_195322.log
-rw-rw-r-- 1 admin admin 24032569 Oct 27 19:57 T_TY_DZ_DZDZ_HZFP_TZD_ZB_DSV_R_Z_195706.log
-rw-rw-r-- 1 admin admin 24032569 Oct 27 19:57 T_TY_DZ_DZDZ_HZFP_TZD_ZB_DSV_R_Z_195726.log
-rw-rw-r-- 1 admin admin 11981146 Oct 27 19:53 T_TY_DZ_DZDZ_QXRZ_TJ_DSV_R_Z_195255.log
-rw-rw-r-- 1 admin admin 11979690 Oct 27 19:53 T_TY_DZ_DZDZ_QXRZ_TJ_DSV_R_Z_195320.log
-rw-rw-r-- 1 admin admin 11981144 Oct 27 19:57 T_TY_DZ_DZDZ_QXRZ_TJ_DSV_R_Z_195706.log
-rw-rw-r-- 1 admin admin 11981143 Oct 27 19:57 T_TY_DZ_DZDZ_QXRZ_TJ_DSV_R_Z_195724.log

[root@201p43112.cloud.p44.a115 /home/admin/version/TY/shell/log/commit/20171027/DZ/failLog]
#pwd
/home/admin/version/TY/shell/log/commit/20171027/DZ/failLog
```

➤ 根据失败原因，再到文档服务器查看源数据文件

```
2017-11-28 01:02:44.744 [0-0-45-reader] INFO UnstructuredStorageReaderUtil - CsvReader使用默认值{"captureRawRecord":true,"columnCount":0,"comment":"#",
rrentRecord":1,"delimiter":",","escapeMode":1,"headerCount":0,"rawRecord":1,"recordDelimiter":"\u0000","safetySwitch":false,"skipEmptyRecords":true,"te
qualifier":"","trimWhitespace":true,"useComments":false,"useTextQualifier":true,"values":{}}]
2017-11-28 01:02:44.748 [0-0-45-reader] INFO FtpReader$Task - end read source files...
2017-11-28 01:02:44.790 [0-0-45-writer] INFO OdpsWriterProxy - write block 0 ok.
2017-11-28 01:02:44.790 [0-0-45-writer] INFO OdpsWriter$Task - Slave which uploadId=[201711280102442ab510560086b363] begin to commit blocks:[
0
].
2017-11-28 01:02:44.946 [0-0-45-writer] INFO OdpsWriter$Task - Slave which uploadId=[201711280102442ab510560086b363] commit blocks ok.
2017-11-28 01:02:44.971 [taskGroup-0] INFO TaskGroupContainer - taskGroup[0] taskId[45] is succeeded, used[301]ms
2017-11-28 01:02:44.974 [taskGroup-0] INFO TaskGroupContainer - taskGroup[0] taskId[46] attemptCount[1] is started
2017-11-28 01:02:45.003 [0-0-46-writer] INFO OdpsWriter$Task - task uploadId=[201711280102442ab510560086b363].
2017-11-28 01:02:45.053 [0-0-46-reader] INFO FtpReader$Task - start read source files...
2017-11-28 01:02:45.053 [0-0-46-reader] INFO FtpReader$Task - reading file : [/home/DSV/FW/20171127/HTJS_CB_FPCQL_MX_QD_2017-11-27_01-50-29_02363_data.d
2017-11-28 01:02:45.055 [0-0-46-reader] INFO UnstructuredStorageReaderUtil - CsvReader使用默认值{"captureRawRecord":true,"columnCount":0,"comment":"#",
rrentRecord":1,"delimiter":",","escapeMode":1,"headerCount":0,"rawRecord":1,"recordDelimiter":"\u0000","safetySwitch":false,"skipEmptyRecords":true,"te
qualifier":"","trimWhitespace":true,"useComments":false,"useTextQualifier":true,"values":{}}]
2017-11-28 01:02:45.058 [0-0-46-reader] INFO FtpReader$Task - end read source files...
2017-11-28 01:02:45.066 [job-0] ERROR JobContainer - 运行scheduler 模式[standalone]出错。
2017-11-28 01:02:45.070 [job-0] ERROR JobContainer - Exception when job run
com.alibaba.datax.common.exception.DataException: Code:[Framework-14], Description:[DataX传输脏数据超过用户预期，该错误通常是由于源端数据存在较多业务脏数
导致，请仔细检查DataX汇报的脏数据日志信息，或者您可以适当调大脏数据阈值。]。 - 脏数据条数检查不通过，限制是[0]条，但实际上捕获了[2]条。
at com.alibaba.datax.common.exception.DataException.asDataException(DataException.java:26) ~[datax-common-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.util.ErrorRecordChecker.checkRecordLimit(ErrorRecordChecker.java:61) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.scheduler.AbstractScheduler.schedule(AbstractScheduler.java:95) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.job.JobContainer.schedule(JobContainer.java:640) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.job.JobContainer.start(JobContainer.java:158) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.Engine.start(Engine.java:96) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.Engine.entry(Engine.java:182) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.Engine.main(Engine.java:215) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
2017-11-28 01:02:45.070 [job-0] INFO StandAloneJobContainerCommunicator - Total 450870 records, 83046948 bytes | Speed 79.20MB/s, 450870 records/s | Err
2 records, 338 bytes | All Task WaitWriterTime 0.139s | All Task WaitReaderTime 72.069s | Percentage 32.82%
2017-11-28 01:02:45.071 [job-0] INFO StandAloneJobContainerCommunicator - Total 0 records, 0 bytes | Speed 0B/s, 0 records/s | Error 0 records, 0 bytes
All Task WaitWriterTime 0.000s | All Task WaitReaderTime 0.000s | Percentage 0.00%
2017-11-28 01:02:45.071 [job-0] ERROR Engine -
经DataX智能分析,该任务最可能的错误原因是:
com.alibaba.datax.common.exception.DataException: Code:[Framework-14], Description:[DataX传输脏数据超过用户预期，该错误通常是由于源端数据存在较多业务脏数
导致，请仔细检查DataX汇报的脏数据日志信息，或者您可以适当调大脏数据阈值。]。 - 脏数据条数检查不通过，限制是[0]条，但实际上捕获了[2]条。
at com.alibaba.datax.common.exception.DataException.asDataException(DataException.java:26) ~[datax-common-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.util.ErrorRecordChecker.checkRecordLimit(ErrorRecordChecker.java:61) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.scheduler.AbstractScheduler.schedule(AbstractScheduler.java:95) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.job.JobContainer.schedule(JobContainer.java:640) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.job.JobContainer.start(JobContainer.java:158) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.Engine.start(Engine.java:96) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
at com.alibaba.datax.core.Engine.entry(Engine.java:182) ~[datax-core-0.0.1-SNAPSHOT.jar:na]
```