广东省国税局

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 | 数据上云小组 | 编写日期 | 2018-03-31 |
| 审核人 | 个人 | 审核日期 | 2018-03-31 |
| 公开范围 | 广东省国税局大数据项目组 | | |
| 建设单位 | 广东省国税局风控处 | | |
| 承建单位 | 北京华云慧通科技有限公司 | | |

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](#_Toc511028661)

[1.1. 目的 4](#_Toc511028662)

[1.2. 范围 4](#_Toc511028663)

[1.3. 定义、业务术语、缩略语 4](#_Toc511028664)

[1.4. 参考资料 4](#_Toc511028665)

[2. DataX+ogg全量上云实施 4](#_Toc511028666)

[2.1. Ogg源端目标端配置流程 4](#_Toc511028667)

[2.1.1. OGG简介 4](#_Toc511028668)

[2.1.1.1. 相关术语 5](#_Toc511028669)

[2.1.1.2. 原理特点 6](#_Toc511028670)

[2.1.1.3. 安装使用 6](#_Toc511028671)

[2.1.1.3.1. 下载安装 6](#_Toc511028672)

[2.1.1.3.2. 双端配置 8](#_Toc511028673)

[2.1.1.3.3. 测试OGG 14](#_Toc511028674)

[2.2. Datax全量初始化实施 16](#_Toc511028675)

[2.2.1. 总体流程及结构 16](#_Toc511028676)

[2.2.1.1. 流程图 16](#_Toc511028677)

[2.2.1.2. 文件目录结构 17](#_Toc511028678)

[2.2.2. 初始化配置 19](#_Toc511028679)

[2.2.2.1. 配置ty\_datasource.conf 20](#_Toc511028680)

[2.2.2.2. 配置ty\_createJson\_ql.conf 21](#_Toc511028681)

[2.2.2.3. odps建表 21](#_Toc511028682)

[2.2.2.3.1. 源表与目标表字段类型映射 21](#_Toc511028683)

[2.2.2.3.2. 生成建表脚本 22](#_Toc511028684)

[2.2.2.3.3. odps建表 24](#_Toc511028685)

[2.2.2.4. 生成Json文件 25](#_Toc511028686)

[2.2.3. 提交任务上云 26](#_Toc511028687)

[2.2.3.1. 统计源表记录数 26](#_Toc511028688)

[2.2.3.2. 提交datax任务 26](#_Toc511028689)

[2.2.4. 结果统计(非必须) 29](#_Toc511028690)

[2.2.5. 合并分区 31](#_Toc511028691)

[2.3. Datax增量初始化实施 35](#_Toc511028692)

[2.3.1. 总体流程及结构 35](#_Toc511028693)

[2.3.1.1. 流程图 35](#_Toc511028694)

[2.3.1.2. 具体步骤 36](#_Toc511028695)

[2.3.1.3. 补数据 42](#_Toc511028696)

[2.3.1.4. 分析失败任务 44](#_Toc511028697)

1. 简介
   1. 目的
   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. 定义、业务术语、缩略语
  2. 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》

1. DataX+ogg全量+增量上云实施
   1. Ogg源端目标端配置流程
      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

* + - 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，发布阀值告警等，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。
  + - 1. 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
* 复制冲突检测和解决
* 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
  + - 1. 安装使用
         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

源端安装

源端配置/response/oggcore.rsp文件



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>

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

* + - * 1. 双端配置

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

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

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 hxnl,tranlog, begin now --如果是9i则需要指定thread 1，thread 2

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

extract hxnl

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.\*;

源端pump配置

* 配置并增加投递进程phxnl

GGSCI> add ext phxnl,exttrailsource ./dirdat/no

GGSCI> add rmttrail ./dirdat/no , extract phxnl

extract phxnl

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.\*;

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

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

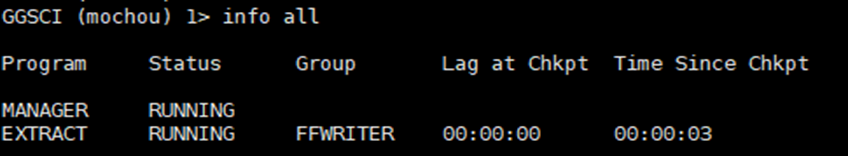
SOURCEDEFS ./dirdef/ogg.def

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



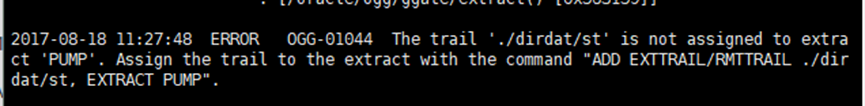
源端进程开启

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



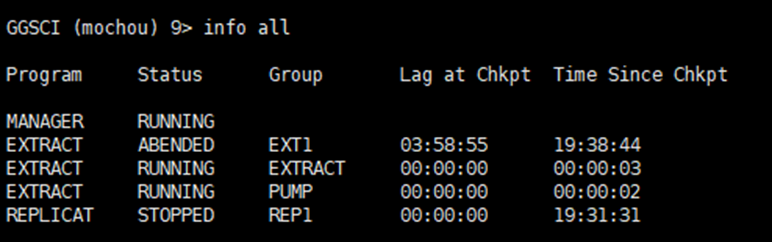
这个报错已经说得很清楚了可以通过下面命令来解决

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

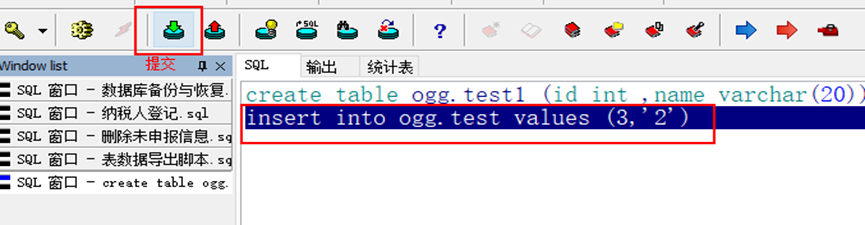
至于ERROR OGG-10668 ABENDING 这个错误是代表着配置有错误，即只要有错误配置这个ERROR就会出现，所以不用管它，只需更改好正确配置，错误标志就会消失

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)

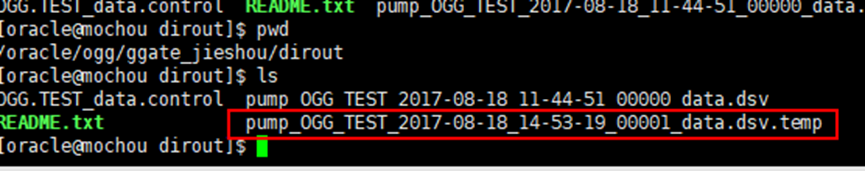
当配置的抽取和投递进程都显示RUNNING时，即代表源端，目标端都已准备好



* + - * 1. 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

* 1. Datax全量初始化实施
     1. 总体流程及结构
        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：



* + - 1. 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

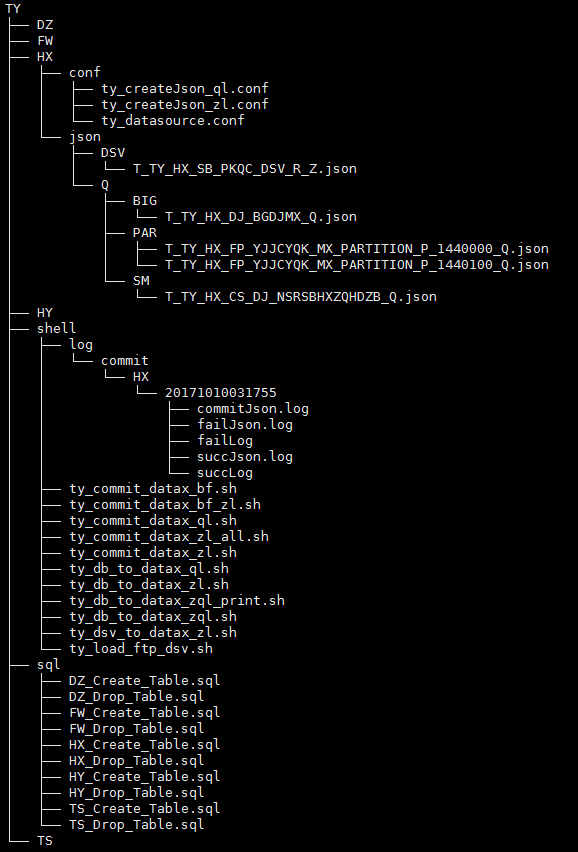
2) shell 所有的shell脚本目录

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



* + 1. 初始化配置

以下路径均以HX为例说明

* + - 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="LTAIwWpd4N89e3d4" #ODPS 的accessid  odpsaccessKey="NfvcI2hcfXNnG4rq8lOllSXQOZtH4L" #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="86.12.72.28:1521/gdsthxff"#连接源库jdbc地址  user="dsjsy"#源库用户名  pass="\*\*\*\*\*\*" #源库密码  odpsJdbc="jdbc:oracle:thin:@${jdbc}"#连接odps的jdbc  ###############################################################################  #文件服务器地址，用于存储增量dsv文件  ###############################################################################  ftpIp="192.168.154.101"##ftp服务器地址  ftpPort="22"##ftp服务器地址  ftpUser="root"##ftp服务器用户名  ftpPasswd="\*\*\*\*\*\*"##ftp服务器密码  ftpBasePath="/home/DSV/HX"  ###############################################################################  #其他配置  ###############################################################################  ##是否开启版本控制  checkVersion=false##开启版本控制(未使用)  ##ogg增量文件后缀  oggSubfix="dsv"  ##ogg增量文件分隔符  spiltFlag="@@"  ##oracle\_home配置  TY\_ORACLE\_HOME="/oradata/app/oracle/product/11.2.0/db\_1"  ##oracle\_lib配置  TY\_LD\_LIBRARY\_PATH="${TY\_ORACLE\_HOME}/lib"  ############################################################################### |

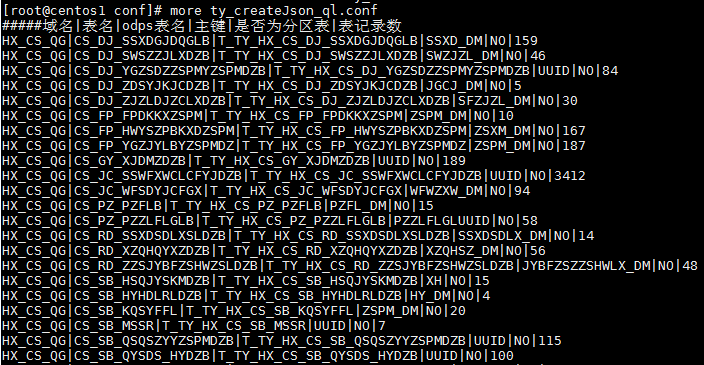
* + - 1. 配置ty\_createJson\_ql.conf

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\_SSXDGJDQGLB|T\_TY\_HX\_CS\_DJ\_SSXDGJDQGLB|SSXD\_DM|NO|159  HX\_CS\_QG|CS\_DJ\_SWSZZJLXDZB|T\_TY\_HX\_CS\_DJ\_SWSZZJLXDZB|SWZJZL\_DM|NO|46 |



* + - 1. odps建表
         1. 源表与目标表字段类型映射

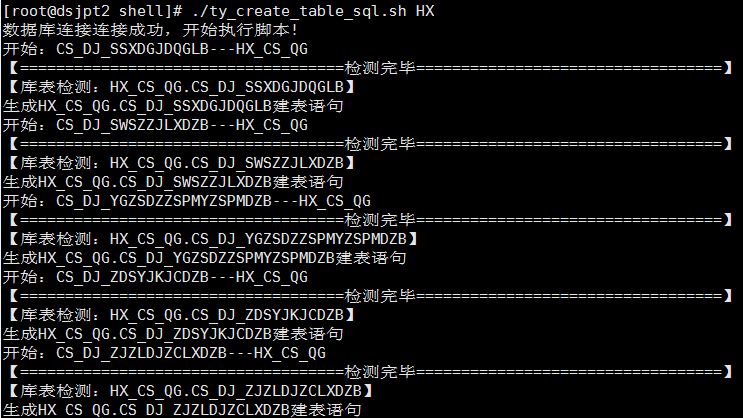
根据odps所支持的字段类型，我们定义oracle与odps的字段类型的映射关系如下



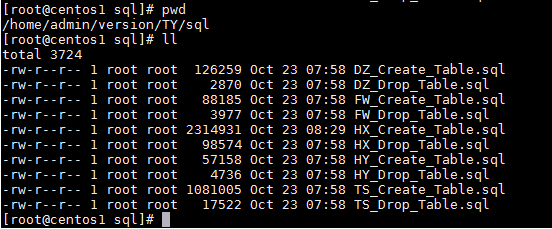
这里shell脚本中已做了映射处理，此步骤无需操作。

* + - * 1. 生成建表脚本

shell目录下执行脚本：./ty\_create\_table\_sql.sh HX



执行完成后，在TY/sql下生成HX\_Create\_Table.sql文件



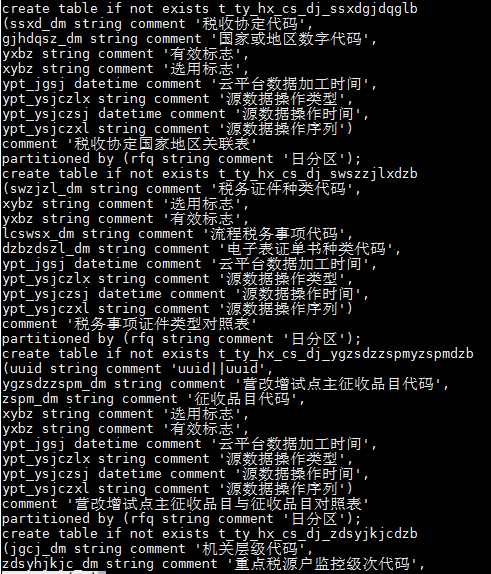
在源表基础上新增了以下字段:

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）

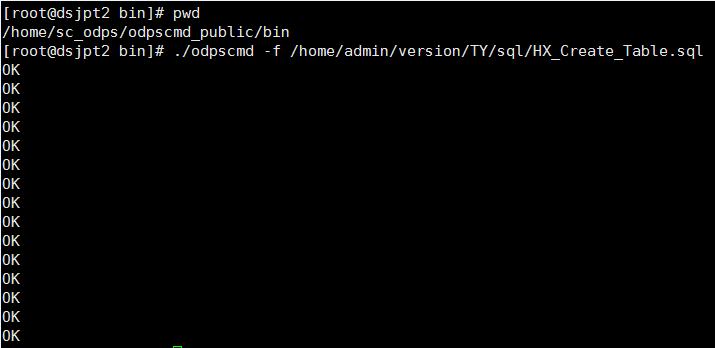
ypt\_ysjczxl（云平台源数据操作序列）



* + - * 1. 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

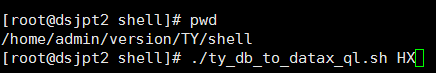


* + - 1. 生成Json文件

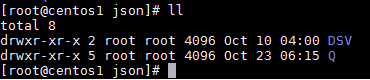
Json文件是指提交datax任务所对应的json文件

shell目录下执行脚本

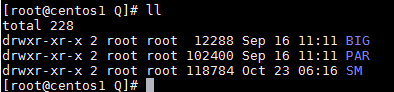
./ty\_db\_to\_datax\_ql.sh HX



执行完成后在HX/json目录下生成Q（全量），其中DSV为增量目录



全量目录下有BIG、PAR、SM，分别表示大表，分区表、小表



Json文件示例：

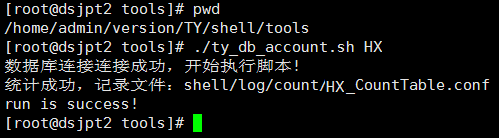


* + 1. 提交任务上云
       1. 统计源表记录数

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

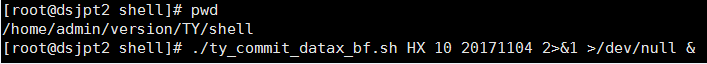
./ty\_db\_account.sh HX



* + - 1. 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 &

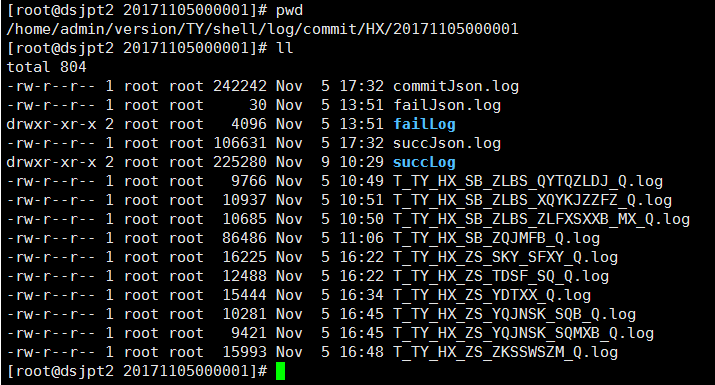


建议通过配置crontab表达式定时在凌晨执行

|  |
| --- |
| 0 0 5 11 \* sh /home/admin/version/TY/shell/ty\_commit\_datax\_bf.sh HX 10 20171104 2>&1 >/dev/null & |

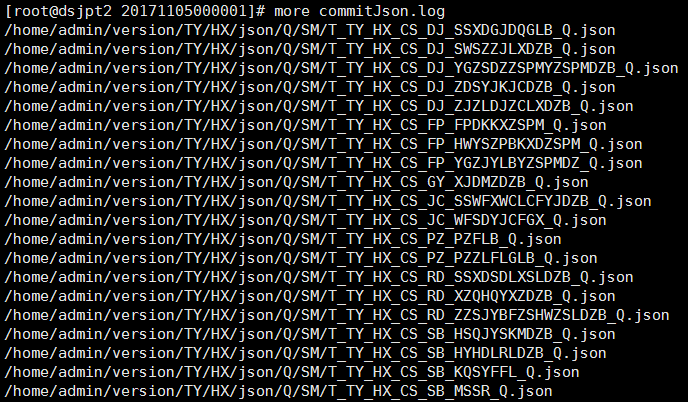
其中参数10为并发数任务数，可根据机器硬件配置适当调整,20171104为全量分区

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



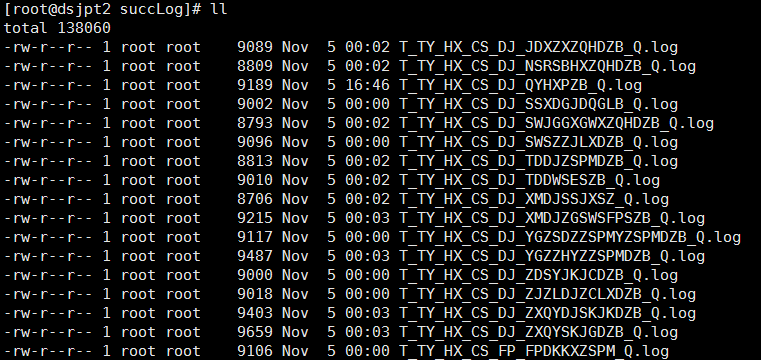
可以看到以下任务指标

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

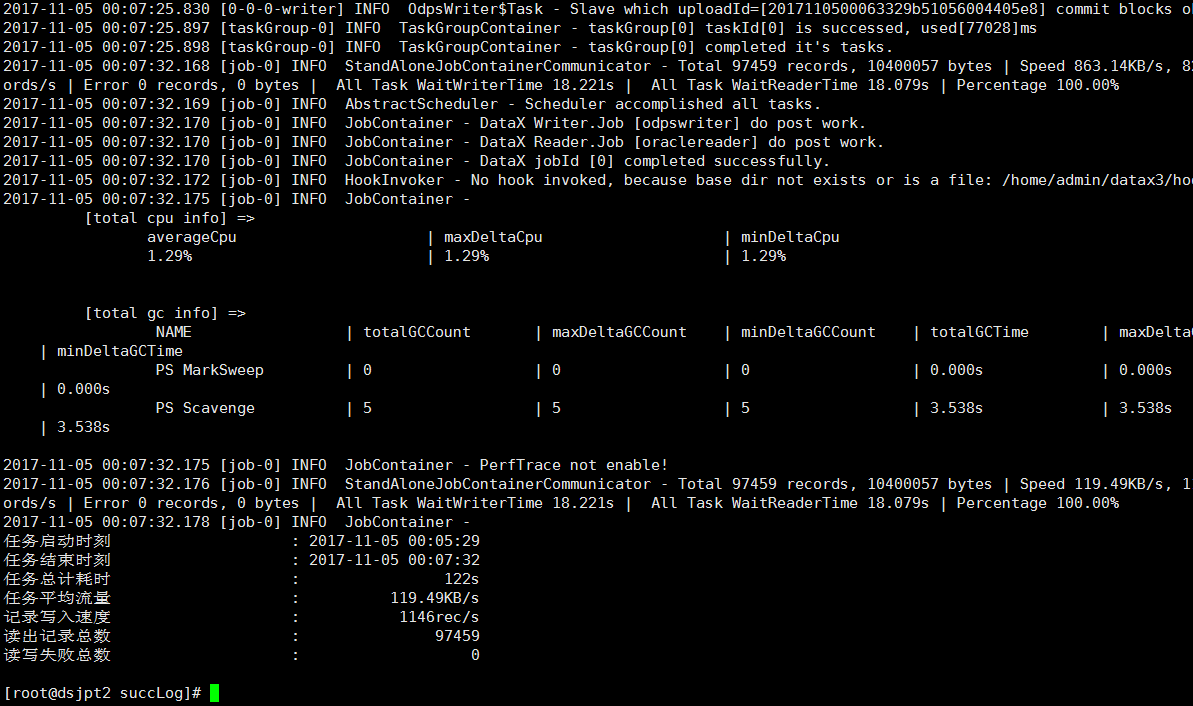


2、运行中：当前目录下的日志文件即为运行中的任务，运行中的任务数量与并发数相同，由图中可以看到正在运行任务数为10

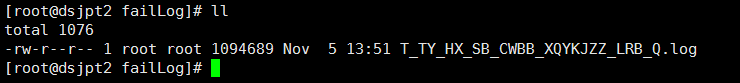
3、成功的任务：在succLog目录下为已运行成功的任务



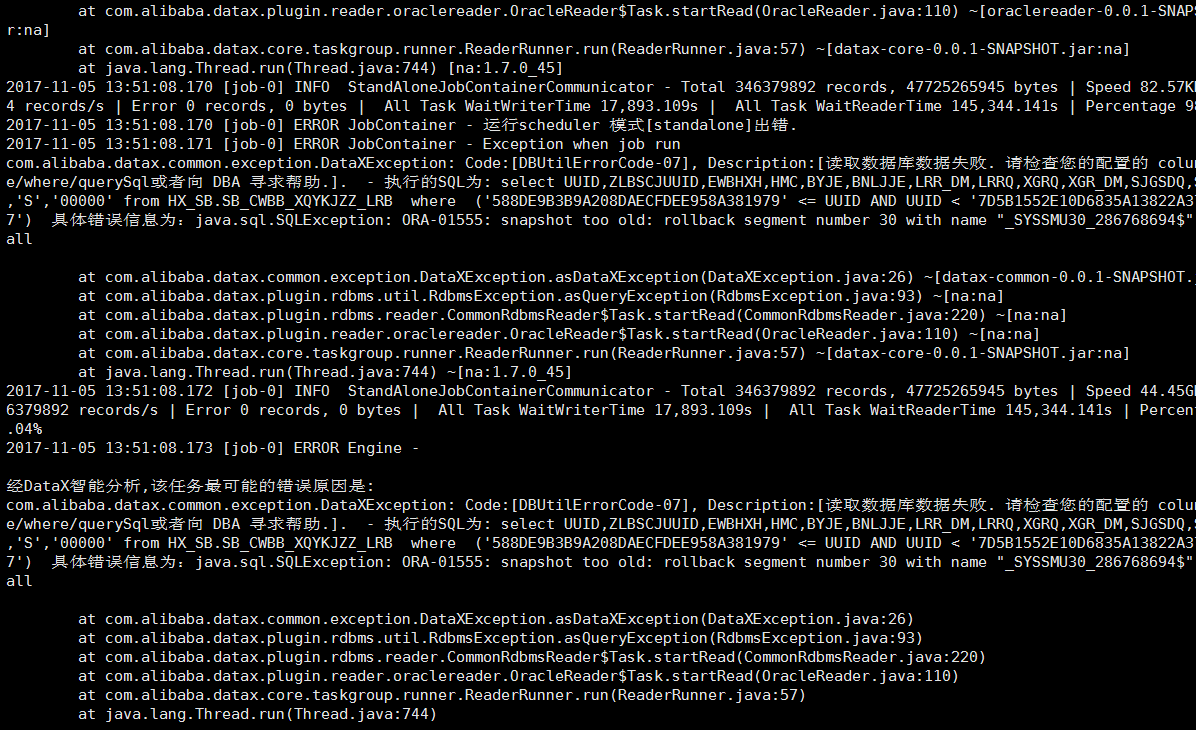
具体会记录任务的读出记录总数、读写失败总数、任务平均流量、任务总计耗时等信息



4、失败的任务：在failLog目录下为已运行失败的任务

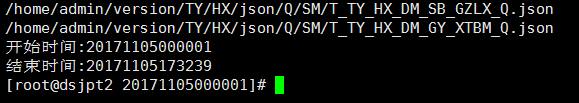


可查看具体的错误日志信息



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。

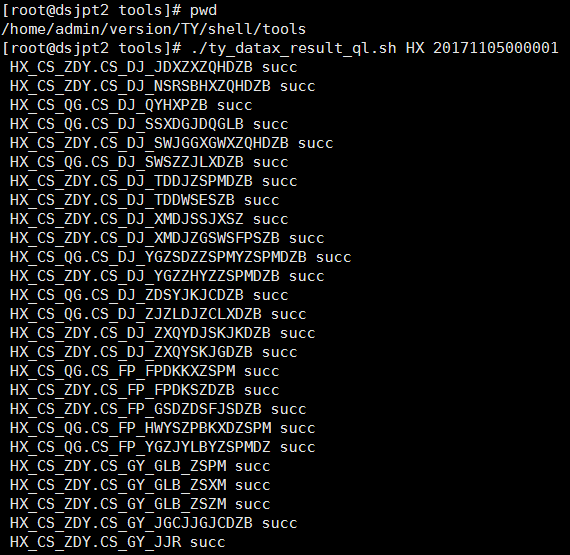


* + 1. 结果统计(非必须)

根据产生的日志信息可统计任务完成情况

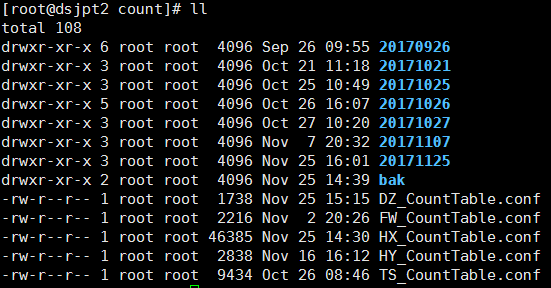
shell/tools下执行脚本

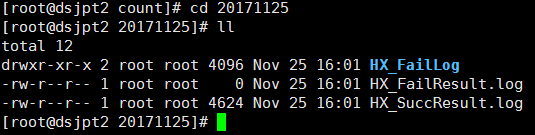
./ty\_datax\_result\_ql.sh HX 日志目录的时间戳

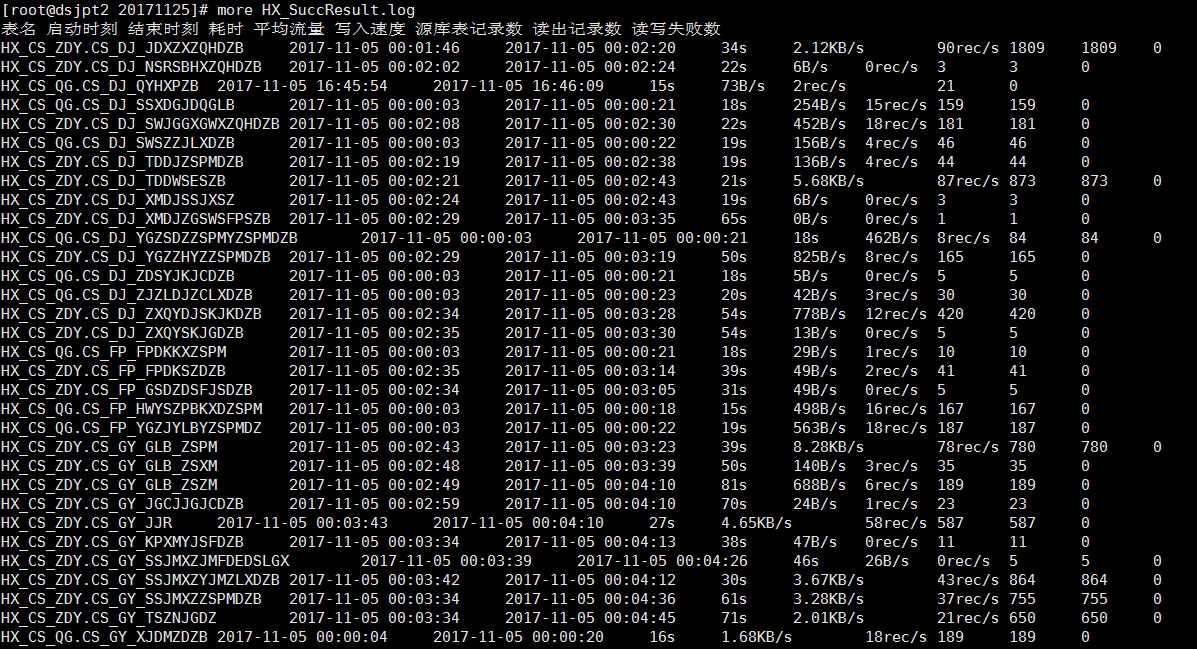


统计结果在目录TY/shell/log/count下

以脚本运行的当天日期为文件夹名称





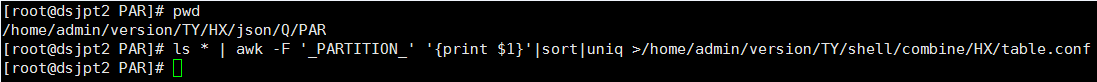


* + 1. 合并分区

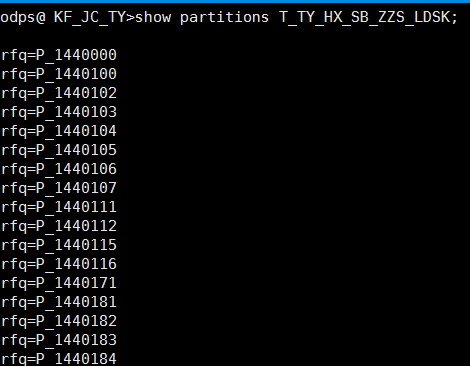
由于分区表是按源表的分区提交，即源表是什么分区，提交到odps就是哪个分区，所以需要将多个分区合并到一个日分区。

1、配置分区表，在TY/HX/json/Q/PAR下执行以下命令

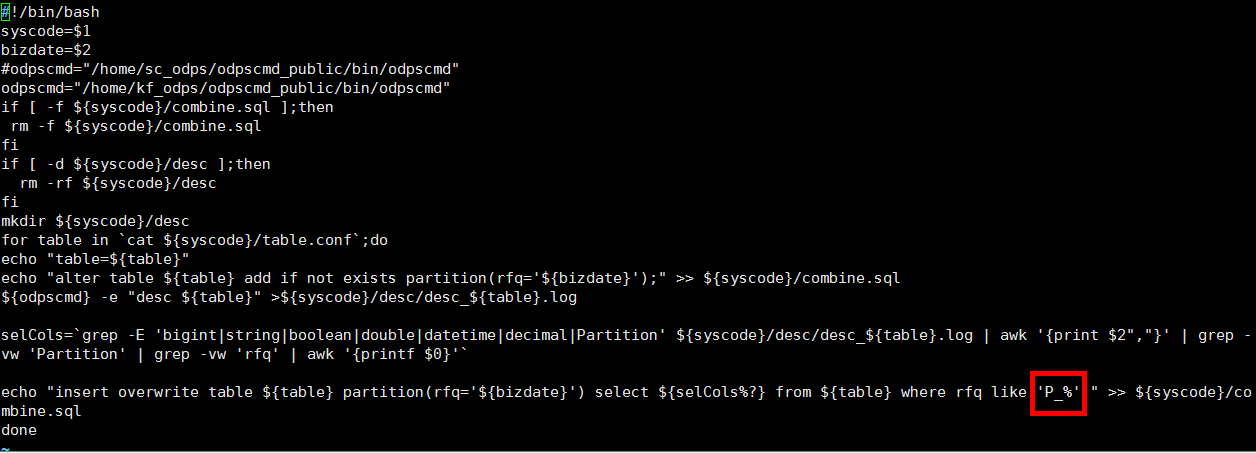
ls \* | awk -F '\_PARTITION\_' '{print $1}'|sort|uniq>/home/admin/version/TY/shell/combine/HX/table.conf



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



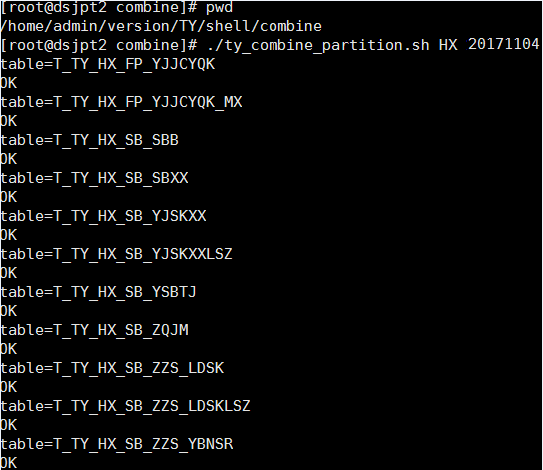
如果不是，请更改脚本shell/combine/ty\_combine\_partition.sh 如下图:rfq likt ‘P\_%’，将P\_改成实际情况，如电子退税则为SYS\_



3、shell/combine下执行脚本

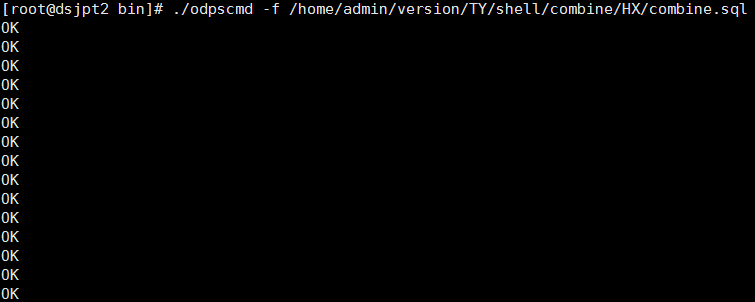
./ty\_combine\_partition.sh HX 20171104

参数20171104为合并到哪个分区



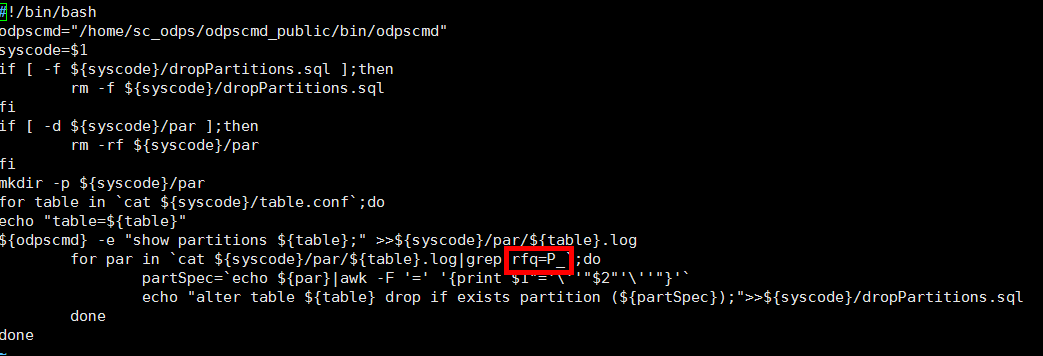
4、在odps客户端执行

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



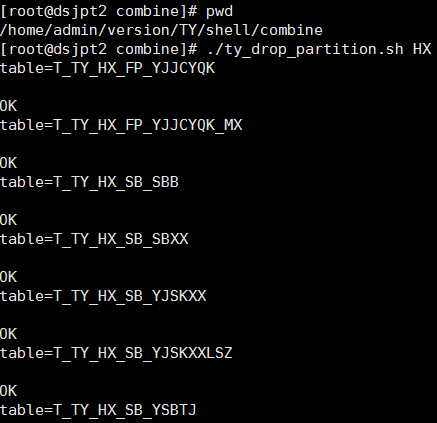
5、删除原有分区

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



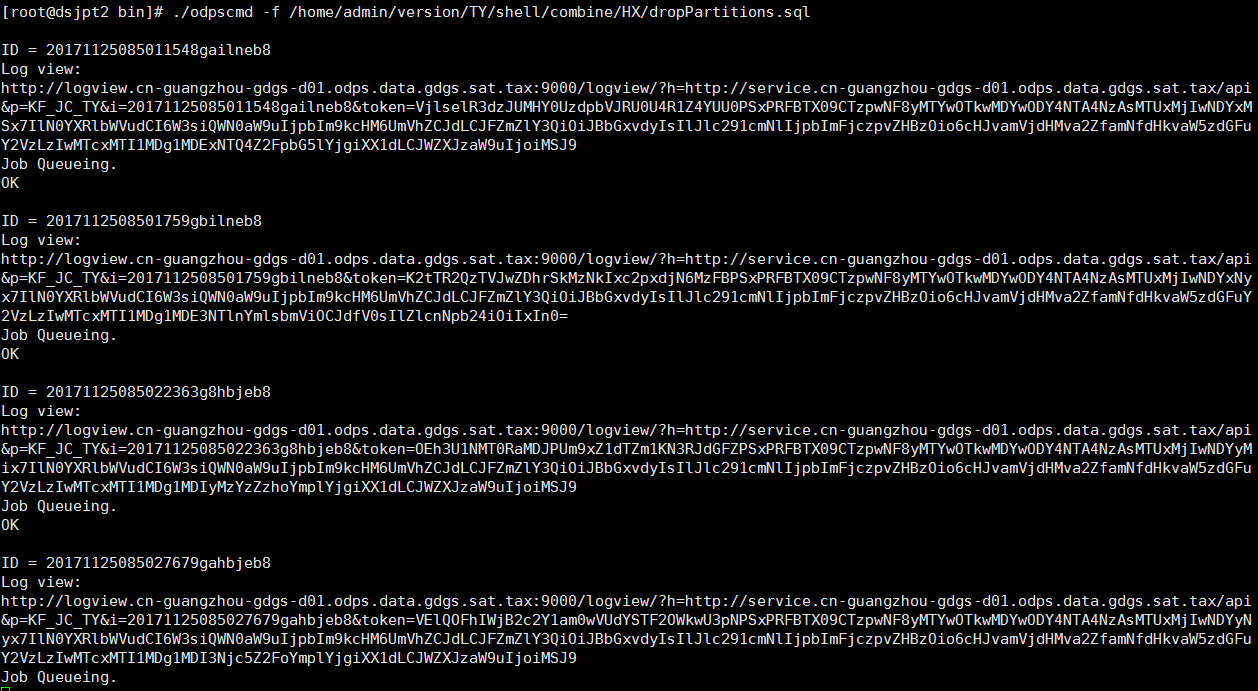
shell/combine下执行脚本

./ty\_drop\_partition.sh HX

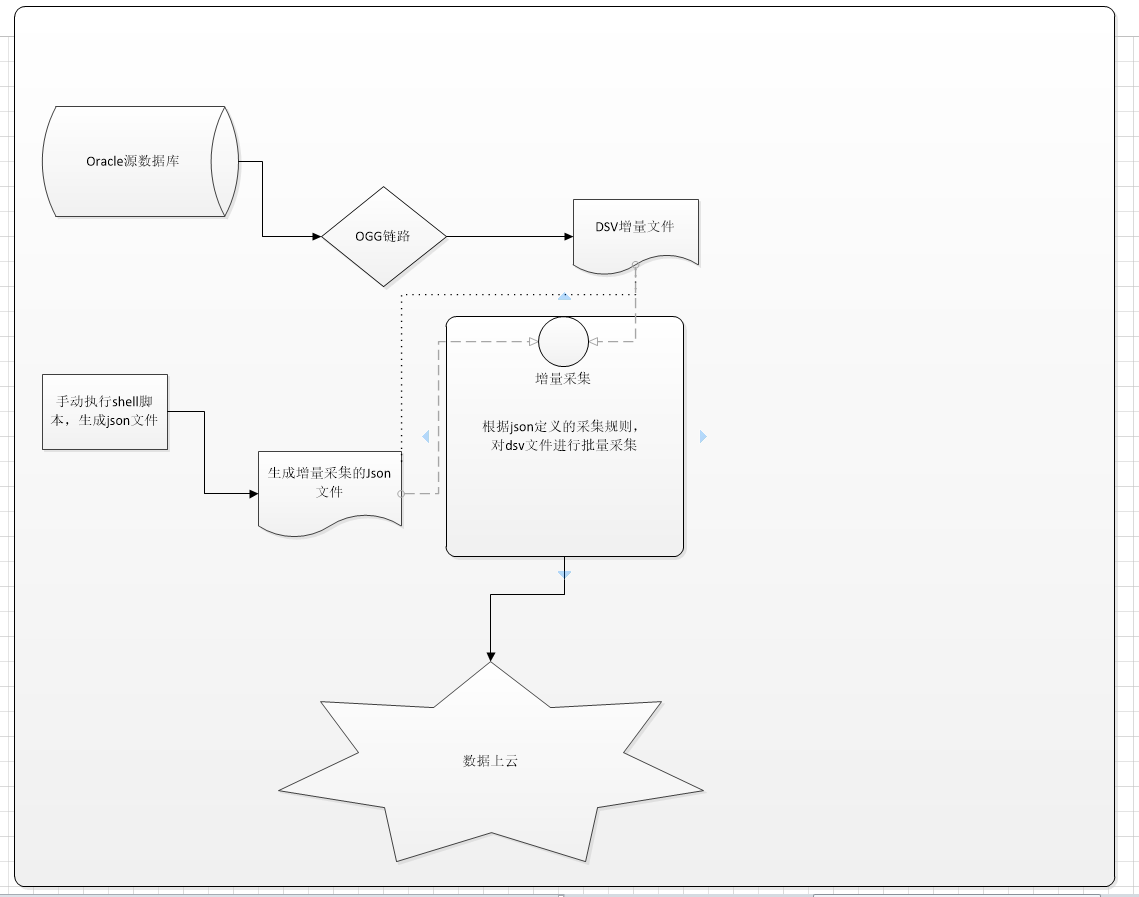


在odps客户端执行

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



* 1. Datax增量初始化实施
     1. 总体流程及结构
        1. 流程图



* + - 1. 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

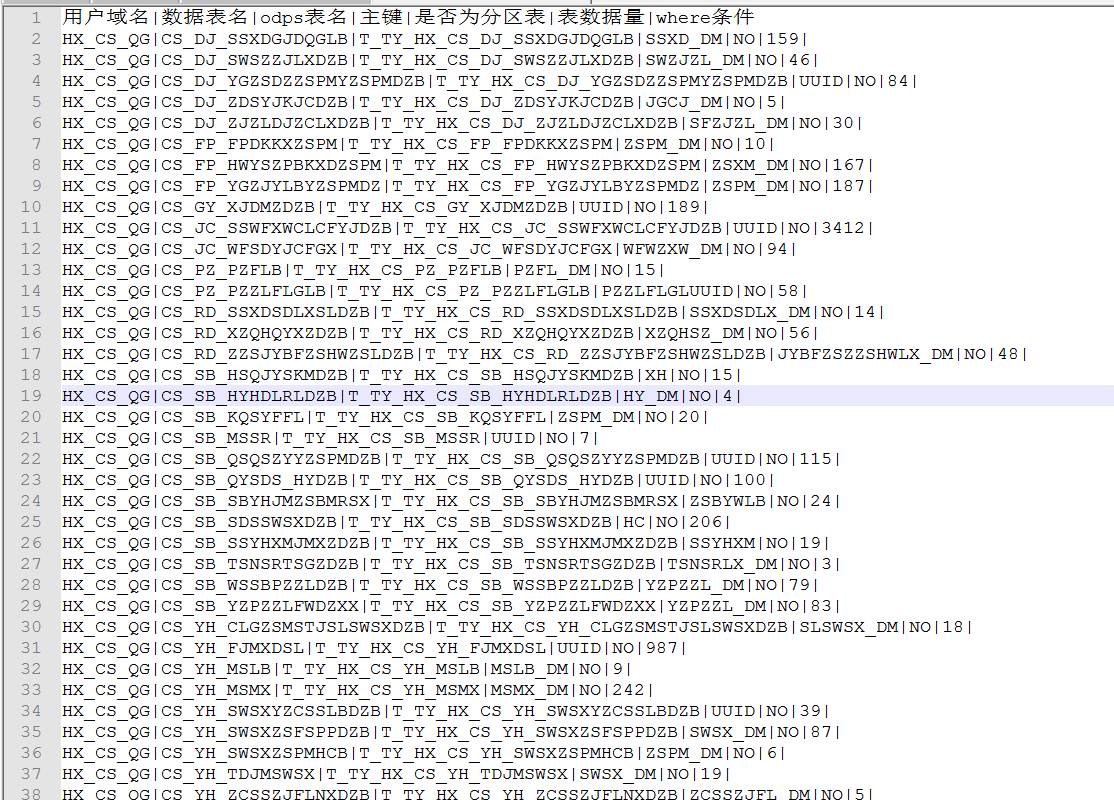
|  |
| --- |
| ###############################################################################  #ODPS 配置 --下面部分都是 ODPS 配置的内容  ###############################################################################  odpsServer="http://service.odps.aliyun.com/api" #ODPS服务器地址  tunnelServer="" #ODPS tunnel地址(未使用)  odpsaccessId="LTAIwWpd4N89e3d4" #ODPS 的accessid  odpsaccessKey="NfvcI2hcfXNnG4rq8lOllSXQOZtH4L" #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="86.12.72.28:1521/gdsthxff"#连接源库jdbc地址  user="dsjsy"#源库用户名  pass="\*\*\*\*\*\*" #源库密码  odpsJdbc="jdbc:oracle:thin:@${jdbc}"#连接odps的jdbc  ###############################################################################  #文件服务器地址，用于存储增量dsv文件  ###############################################################################  ftpIp="192.168.154.101"##ftp服务器地址  ftpPort="22"##ftp服务器地址  ftpUser="root"##ftp服务器用户名  ftpPasswd="\*\*\*\*\*\*"##ftp服务器密码  ftpBasePath="/home/DSV/HX"  ###############################################################################  #其他配置  ###############################################################################  ##是否开启版本控制  checkVersion=false##开启版本控制(未使用)  ##ogg增量文件后缀  oggSubfix="dsv"  ##ogg增量文件分隔符  spiltFlag="@@"  ##oracle\_home配置  TY\_ORACLE\_HOME="/oradata/app/oracle/product/11.2.0/db\_1"  ##oracle\_lib配置  TY\_LD\_LIBRARY\_PATH="${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86.16.29.175上可通过修改来源库连接信息复用同个ty\_datasource配置文件。

1. 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\_SSXDGJDQGLB|T\_TY\_HX\_CS\_DJ\_SSXDGJDQGLB|SSXD\_DM|NO|159|  HX\_CS\_QG|CS\_DJ\_SWSZZJLXDZB|T\_TY\_HX\_CS\_DJ\_SWSZZJLXDZB|SWZJZL\_DM|NO|46| |



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。

1. 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。
2. 拷贝第四步骤的执行脚本和所有json到两台getway机器上，参考命令:

scp –r ty\_commit\_datax\_zl.sh [root@86.16.180.144:/home/admin/vesion/TY/shell](mailto:root@86.16.180.144:/home/admin/vesion/TY/shell)

scp –r ty\_commit\_datax\_zl.sh root@86.16.180.145:/home/admin/vesion/TY/shell

scp –r DSV/\* root@86.16.180.144:/home/admin/vesion/TY/HX/json/DSV

scp –r DSV/\* root@86.16.180.145:/home/admin/vesion/TY/HX/json/DSV

1. 配置Excel增量任务，拷贝该Excel到86.16.196.31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。



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 |  |

1. 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：

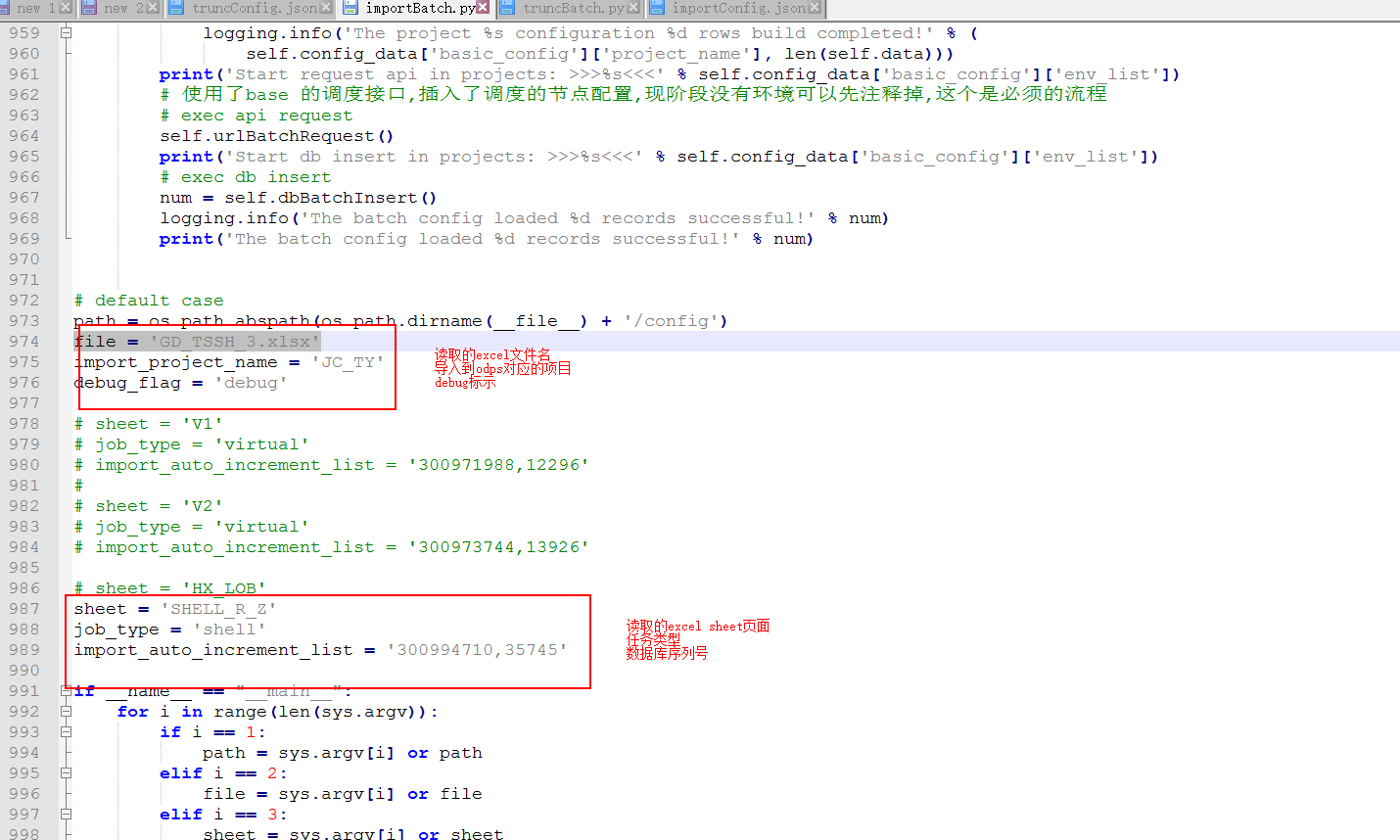
* 对于importBatch.py脚本，需要对以下红框截图内容进行修改，参考如下：

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 86.16.180.144 密码：AliOS%1688 |
| 登录mysql数据库 | mysql -hdpbizide.mysql.data.gdgs.sat.tax -udpbizide -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; |

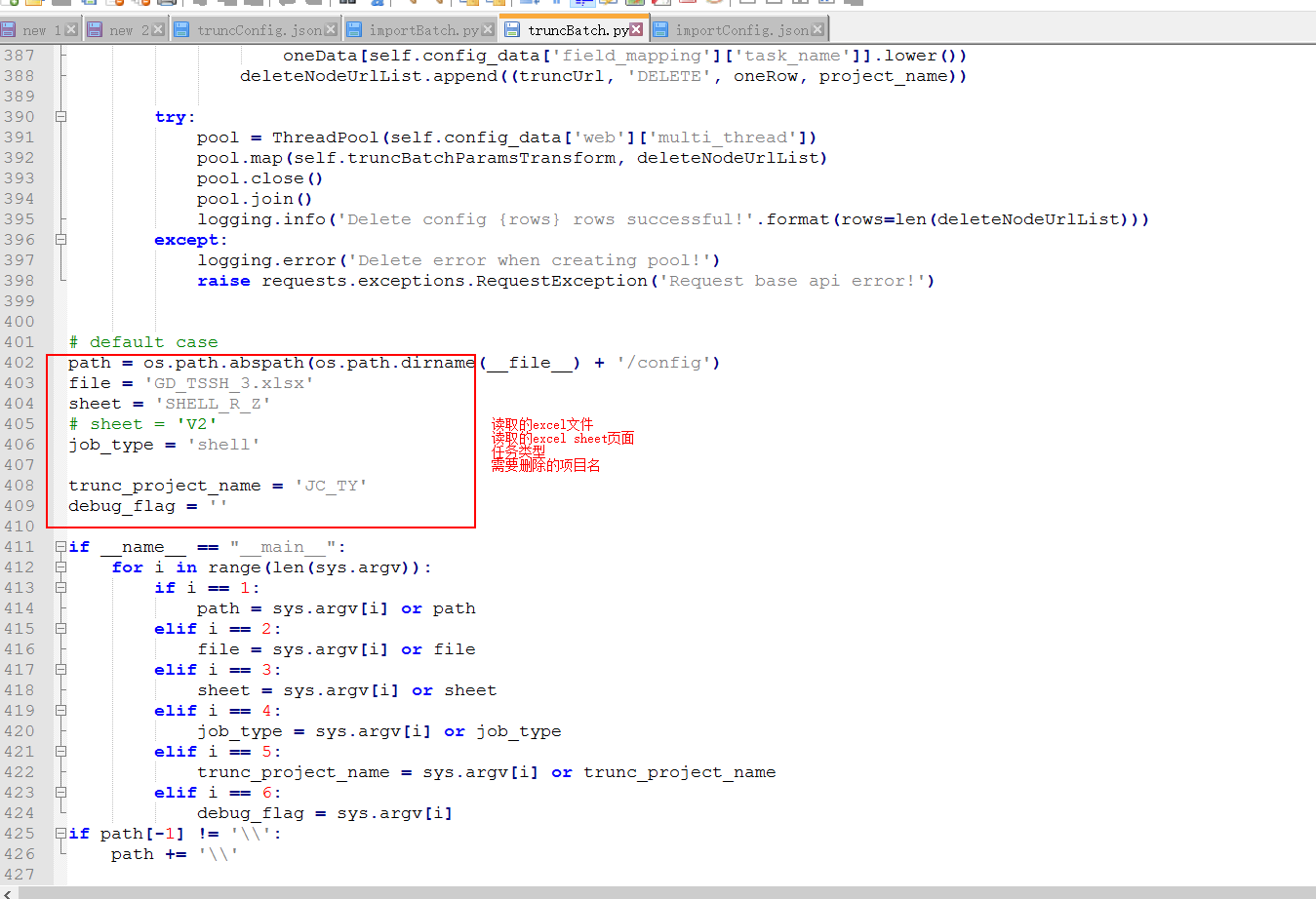
* 对于importBatch.py脚本，需要对以下红框截图内容进行修改，参考如下：

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



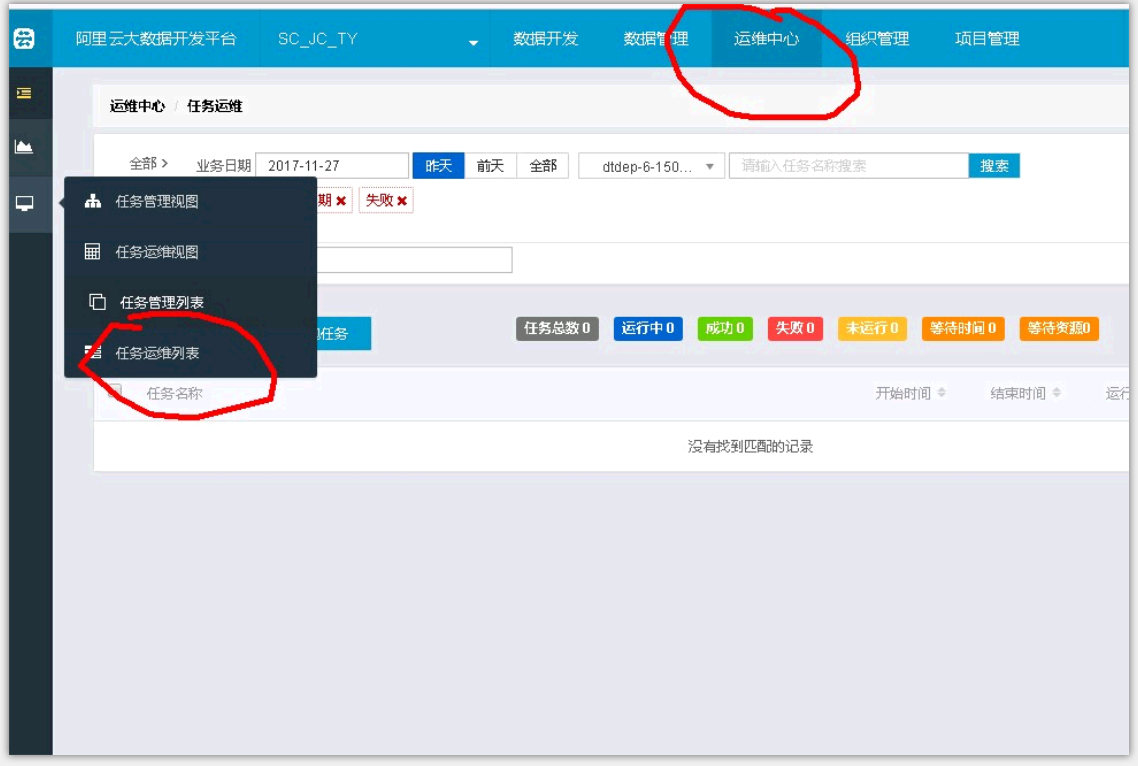
* 对于truncBatch.py脚本，需要对以下红框截图内容进行修改，参考如下：

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

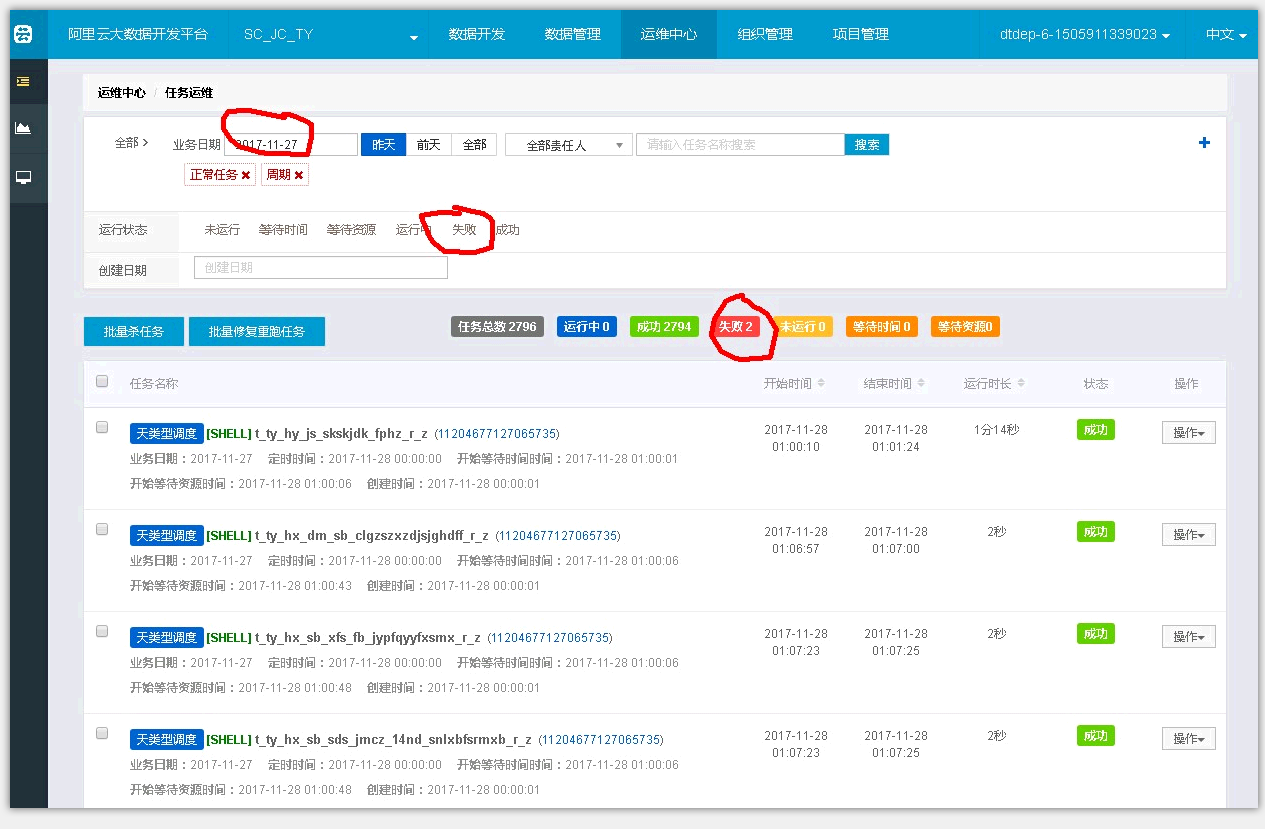


1. 在阿里云大数据开发平台运维增量采集任务，查看是否成功生成了增量任务

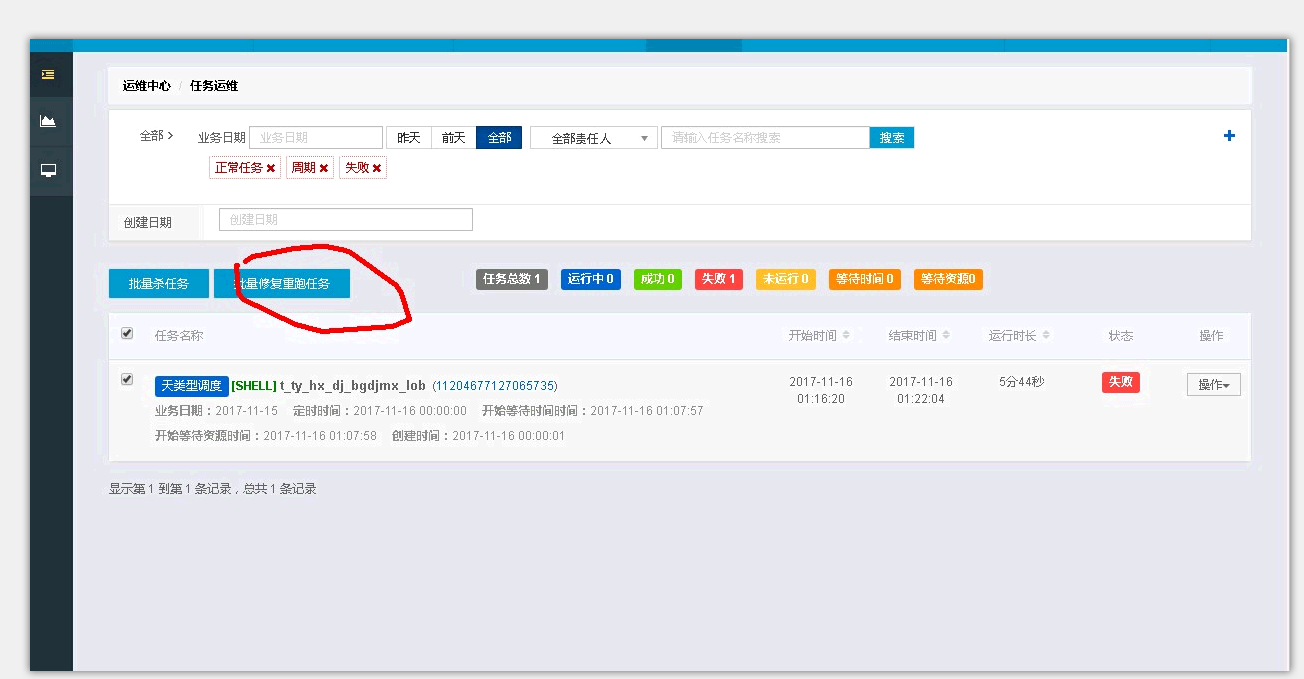
* 登录阿里云大数据管理平台，如下图所示进入运维中心查看增量任务执行情况，操作步骤是：点击运维中心选项卡，进入任务运维列表。



* 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。



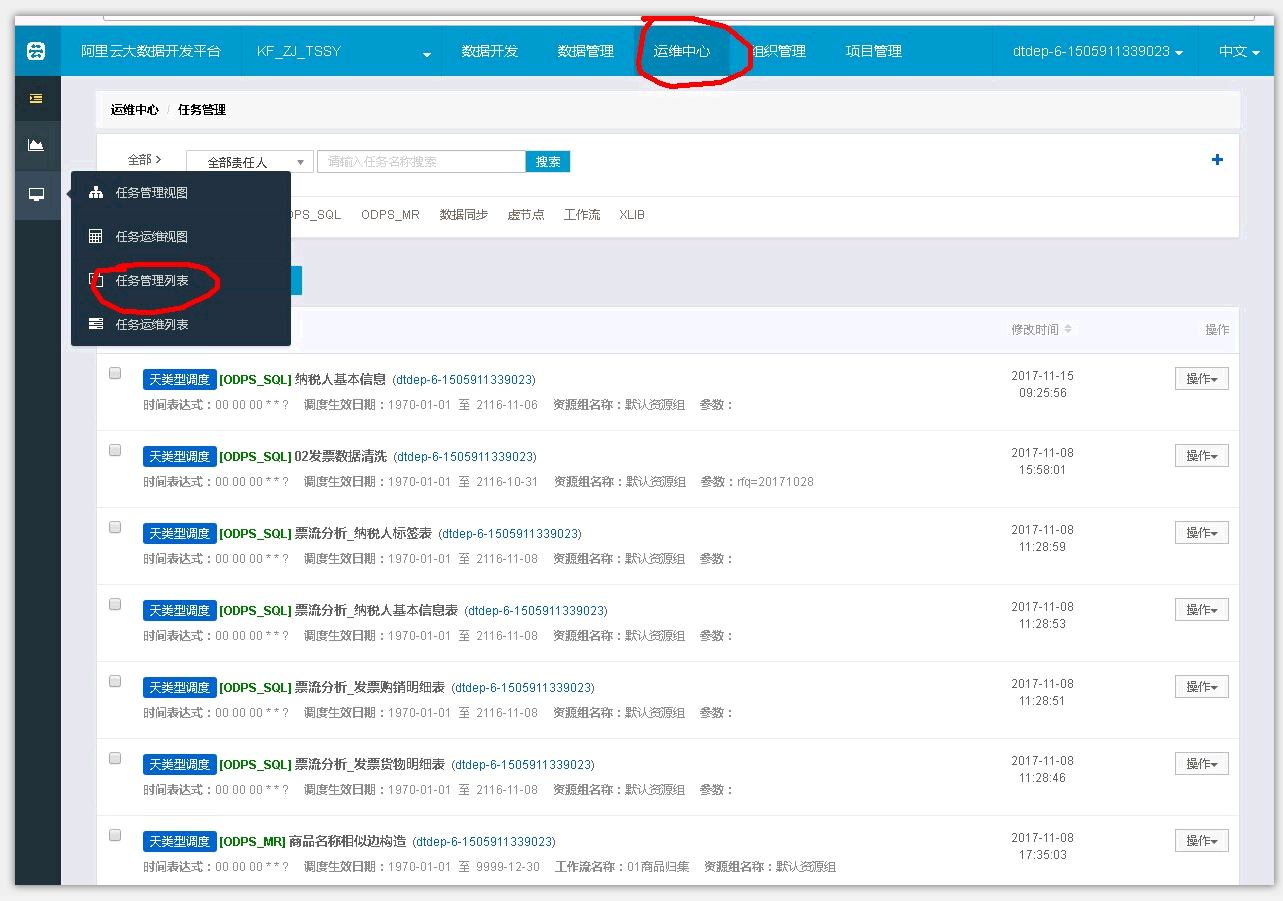
* 批量选中失败任务，直接选中批量修复任务，对任务进行重跑。



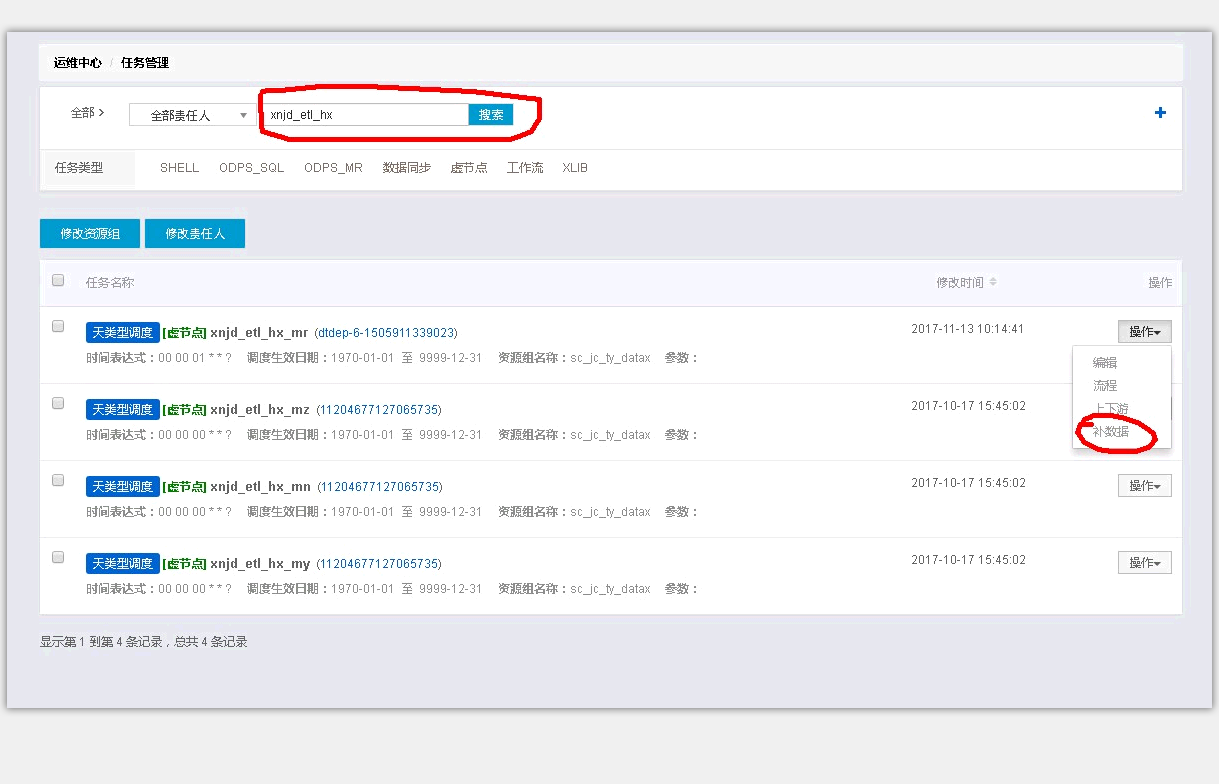
* + - 1. 补数据

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：

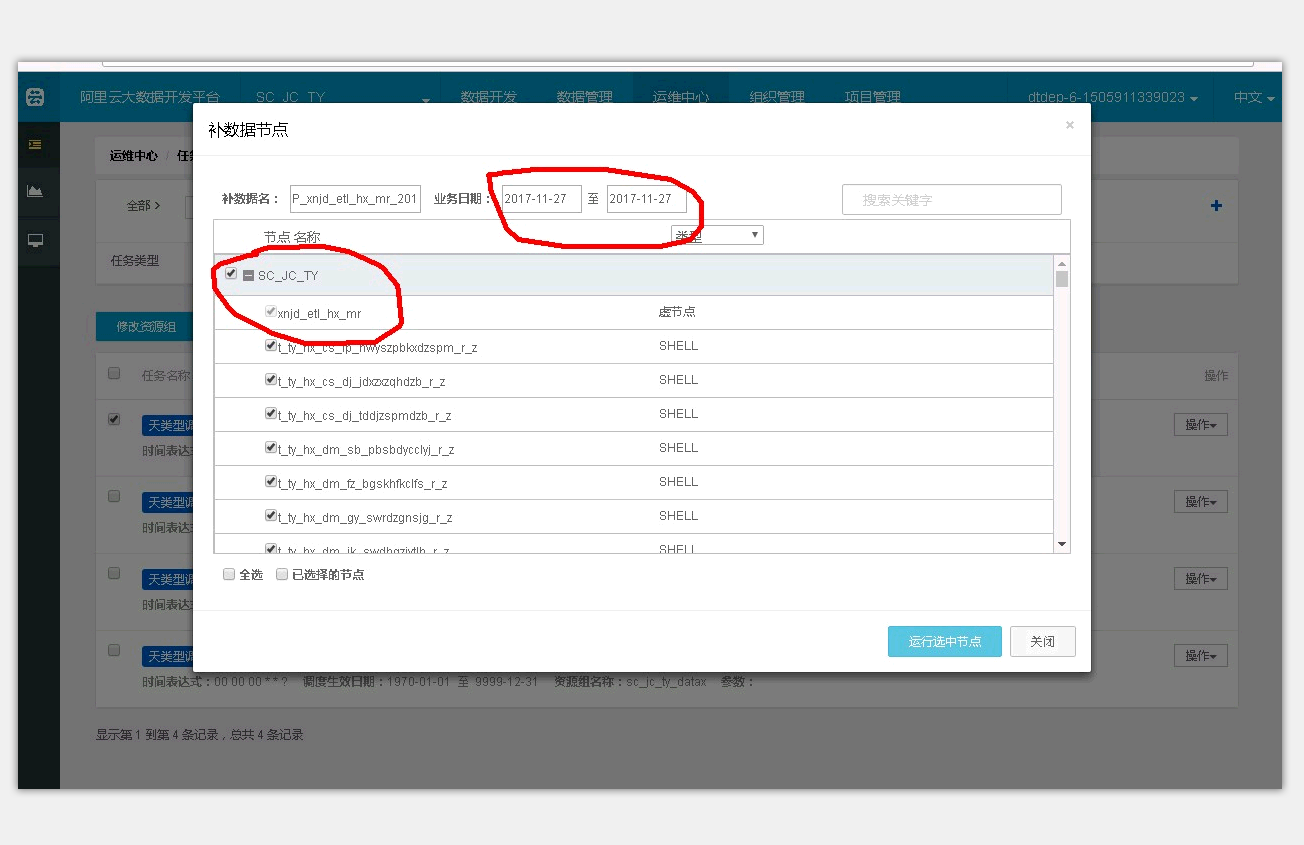
* 打开运维中心，选择任务管理列表，通过搜索关键字查询出需要补数据的任务



* 打开操作下拉框，选中补数据



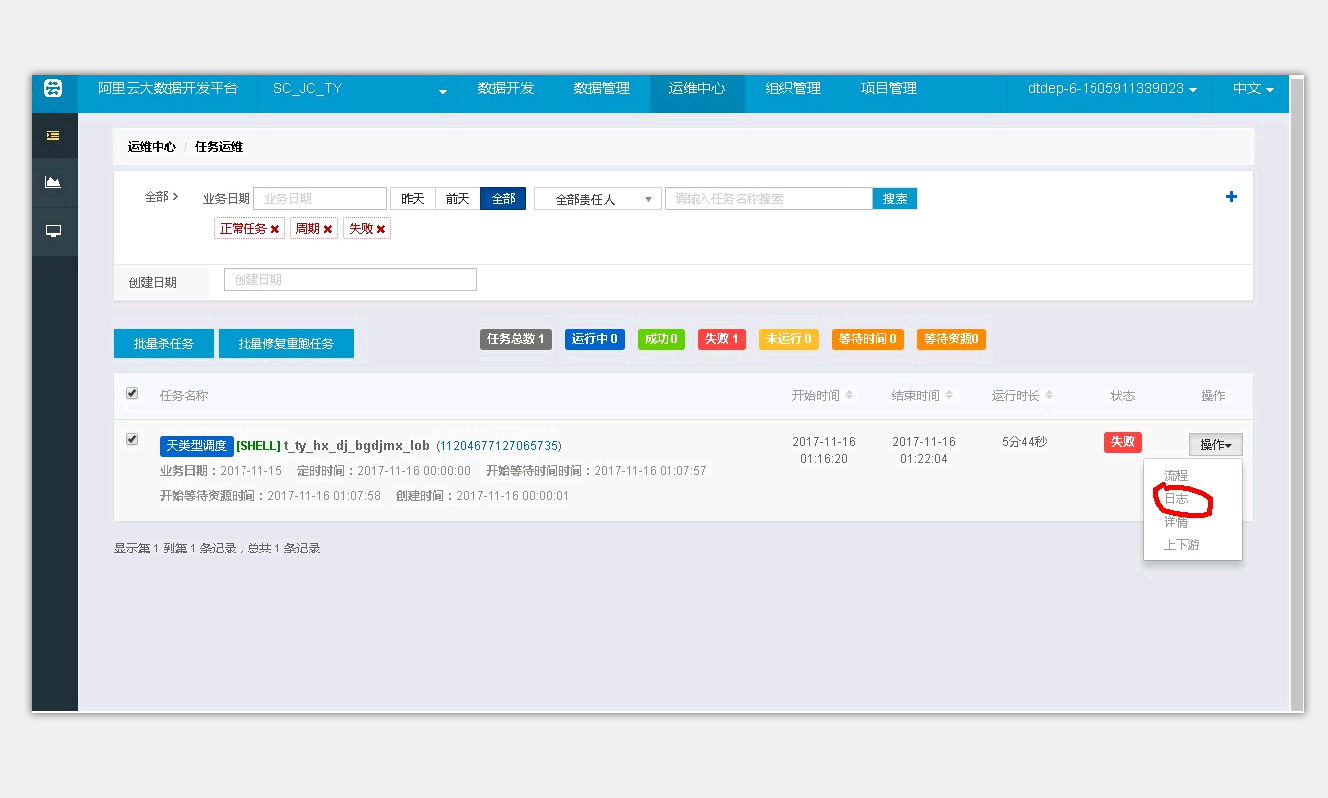
* 选中需要补数据的业务日期，勾选需要补数据的任务列表，然后点击运行选中节点按钮开始补数据



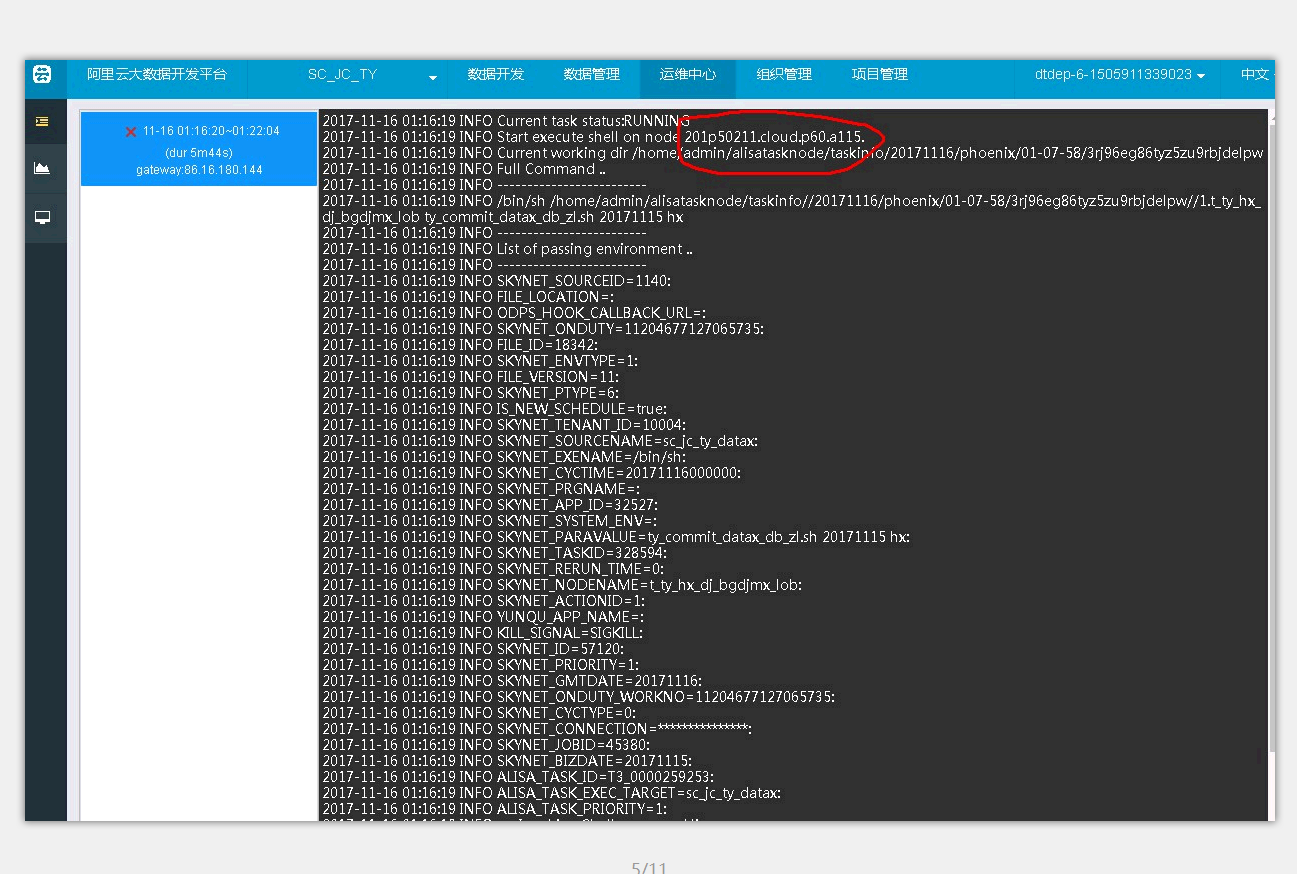
* + - 1. 分析失败任务

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：

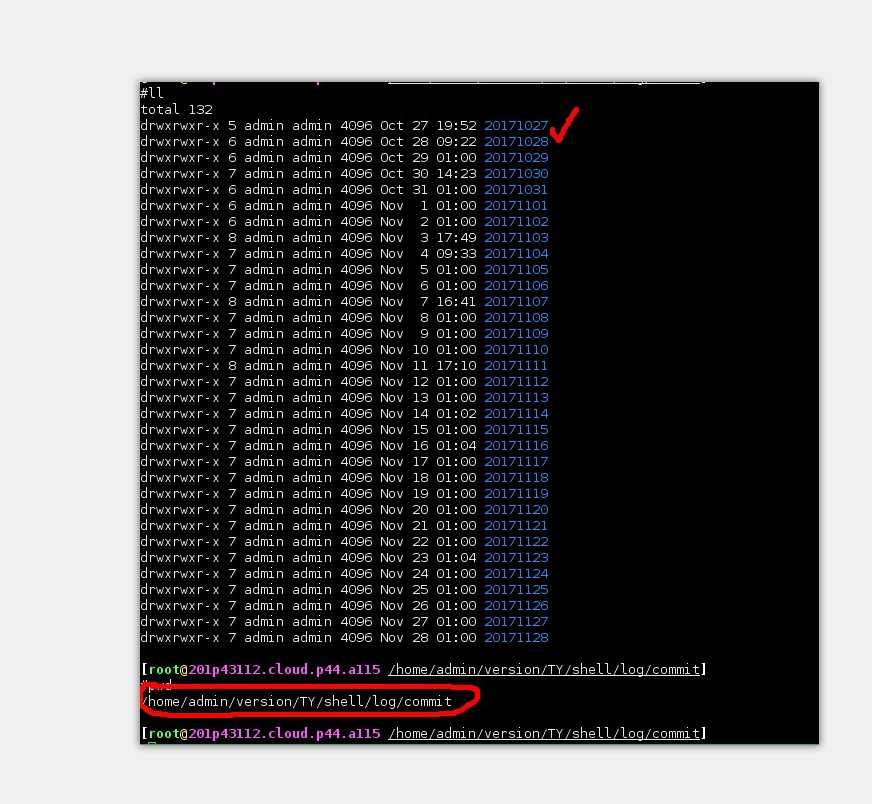
* 在运维中心的任务运维列表中搜索出当天执行失败的任务，点开操作下拉框，点击日志：



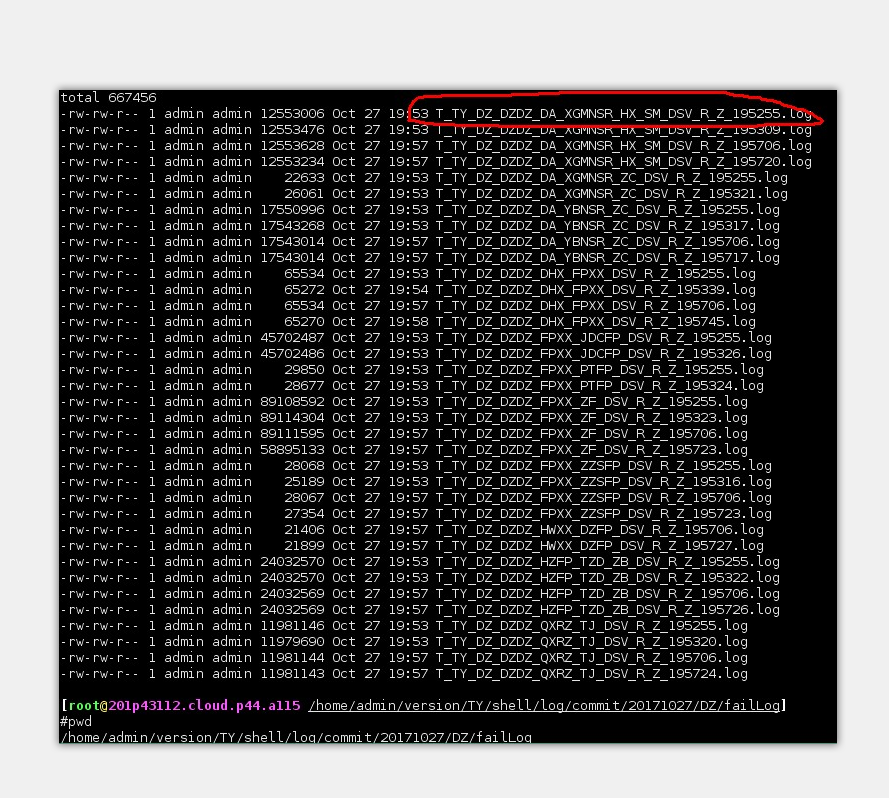
* 在新开的页面中，注意下图红框位置的日志，此处说明了任务的执行节点位于201p50211.cloud.p60.a115机器，通过xshell远程登录该机器



* 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



* 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



* 根据失败原因，再到文档服务器查看源数据文件

