数据迁移自动化脚本

2018-2-20~5-07

1. 环境部署
   1. 安装python 2.7
   2. 安装数据库插件

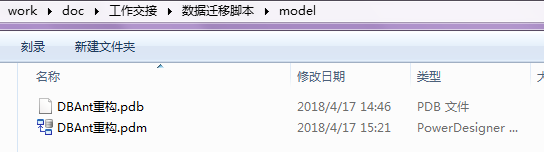


* 1. 安装MySQL数据库

mysql-5.6.22-linux-glibc2.5-x86\_64.tar.gz

* 1. 数据库设计

详见：



1. 脚本说明

## 2.1 现状描述

（1）目前数据迁移使用的是本地安装的MySQL数据库,root口令为空，数据迁移脚本使用的数据库名为dbant。跑几个月没问题，数据库可以迁到线上环境。

（2）目前脚本工作方式，以用户为单位，每个用户下有多张分区表，每一张分区表是一个子任务。

不同用户对应不同的目标库。

（3）每月1日跑一个用户的任务，2号跑一个用户下的任务，依次往后排。通过crontab触发。

（4）目前迁移脚本所有的功能已经进行了大量的自测，已知的BUG均以修复。

（5）10.120.12.253历史库安装了两个Oracle实例，hist1/utf8,hist2/gbk。

hist1/utf8：规定用于存放db4:clic线上数据归档。

hist2/gbk：规定用于存放xhx:tcsvbs.tcsbns线上数据归档。

（6）10.120.12.254 历史库存放的主要是火凤凰的历史数据，目前数据迁移脚本未部署在该服务器上，现阶段还是10.120.12.253上运行，建议等程序稳定运行二个月后，再进行部署。

（7）目前数据迁移程序部署位置/ops/python，需换切至Oracle用户来运行db\_mgt.py程序。

|  |  |
| --- | --- |
| su – oracle  cd /ops/python  ./db\_mgt.py |  |

（8）源程序db\_mgt.py中需要注意的地方

源程序在10.120.12.253服务器部署路径：/ops/python

源程序db\_mgt.py中，函数get\_db中获取dbant数据源参数目前是本地MySQL数据库地址，正式上线需要改为线上dbant环境参数。本地MySQL，root口令为空。dbant用户口令为:dbant

函数read\_para中puppet用户的口令也写死了，该口令配置源在dbant项目中的配置文件中，该口令上线后最好改为从git中动态获取。

文件路径：dbant/apps/base\_config/settings.py，key：ORACLE\_PUB\_USER\_PWD

函数:check()中，检测分区表中所有索引必须为本地索引才进行迁移目前是注释状态，后续为了安全，还是建议启用这个检测。

函数：truncate\_source\_part，清理生产库分区是注释状态，截止目前这个功能已经测试完成，可以放开注释了。

函数：truncate\_dest\_part，清理历史库分区是注释状态，截止目前这个功能已经测试完成，可以放开注释了。

函数：main中，get\_configs(curren\_day('test'))，上线前需要改为：get\_configs(curren\_day())

db\_mgt.py程序定时任务尽量放在晚上在20：00以后运行。

## 2.2 函数描述

以下函数可能不是最新，详见最新代码，都有详细注释

|  |  |
| --- | --- |
| 函数名 | 描述 |
| get\_db | 根据不同类型返回数据库连接对象 |
| read\_para | 根据db\_mgt\_tab\_config.id 返回当前任务配置文件 |
| print\_para | 打印当前任务配置文件，用于调试 |
| get\_logid | 获取当前任务当前日志ID |
| write\_log | 写任务日志 |
| check\_task\_exist | 检测当前任务是否存在 |
| get\_task\_id | 获取当前任务ID |
| write\_task | 写任务表 |
| get\_source\_tablespace | 获取当前任务源库的表空间列表 |
| check\_dest\_tablespace | 检测当前任务源库表空间对应的目标库表空间是否存在 |
| check\_dest\_user\_exist | 检测目标库用户是否存在 |
| check\_dest\_table\_exist | 检测目标库表是否存在 |
| check\_source\_tab\_exist | 检测源库中的表是否存在 |
| check\_source\_tab\_part | 检测源库中的表是否为分区表 |
| check\_dest\_dblink | 检测目标库puppet用户下是否存在相应数据库链路 |
| check\_dest\_dumpdir | 检测目标库puppet用户下是否存在相应目录对象 |
| check\_dumpdir\_exist | 检测目标库puppet用户下是的目录对象指向的路径是否有效 |
| check\_dest\_tab\_part | 检测目标库业务用户中的分区表中特定分区中是否存在数据 |
| get\_source\_index\_names | 获取子任务中的分区表中的所有索引名列表，以逗号分隔 |
| check\_source\_local\_index | 检测源库分区表中的索引是否全部为本地索引 |
| check\_source\_active\_trans | 检测源库分区表中本次迁移的分区上是否存在活动事务 |
| check\_dest\_active\_trans | 检测目标库分区表中本次迁移的分区上是否存在活动事务 |
| check\_source\_part\_grants | 源库分区表是否存在授权引用的用户 |
| get\_source\_charset | 获取源库字符集 |
| get\_dest\_charset | 获取目标库字符集 |
| check\_source\_dest\_charset | 检测源库字符与目标库字符集是否一致 |
| check\_dest\_index\_status | 获取目标库当前任务中的分区表索引是否有效 |
| check | 检测主函数 |
| current\_rq | 获取当前日期 |
| curren\_day | 获取服务器当前天数 |
| read\_exp\_cfg | 获取导出导入参数文件中键对应的值 |
| read\_exp\_content | 获取导出导入日志明细 |
| get\_real\_path | 获取目标库中目录对象指向的路径名 |
| write\_log\_ext | 写任务的扩展日志信息 |
| get\_part\_names | 获取子任务中需要导出的分区名称，以逗号分隔 |
| get\_part\_names\_fmt | 获取子任务中需要导出的分区名称 |
| get\_part\_size | 获取子任务中需要导出的所有子分区占用的空间,单位MB |
| get\_part\_records | 获取当前任务分区表中已迁移完成的分区数 |
| get\_dest\_part\_max\_month | 获取当前任务目标库中已迁移完成的最大分区月份 |
| get\_dest\_part\_records | 获取当前任务目标库中待清理的分区数量 |
| get\_dest\_part\_names | 获取当前任务目标库中待清理的分区列表，以逗号分隔 |
| get\_exp\_part\_names | 获取待导出的分区名列表 |
| get\_exp\_part\_months | 获取待导出的分区月范围 |
| check\_exp\_file\_exist | 检测某一天某一子任务是否已导出数据 |
| exp | 从源库导出特定分区数据至目标库 |
| imp | 将DMP文件导入目标库中 |
| validate | 验证源库目标库分区记录数 |
| truncate\_source\_part | 清除源库业务用户下已导入目标库中的分区数据 |
| truncate\_dest\_part | 清除目标库业务用户下分区数据 |
| upd\_dbant\_cfg | 更新数据迁移配置表信息 |
| get\_configs | 根据数据迁移配置生成待迁移子任务列表 |
| get\_source\_part\_col | 返回源库当前任务中分区表的分区列名 |
| get\_source\_part\_col\_type | 返回源库当前任务中分区表的分区列类型 |
| get\_source\_tab\_part\_list | 返回源库当前任务分区表所有分区名列表 |
| check\_source\_part\_multi\_month | 检测源库分区表中某一个分区中是否存在多个月数据 |
| check\_source\_part\_valid | 检测源库分区表中分区数据是否合法 |
| start\_migration | 数据迁移处理主方法 |
| main | 数据迁移主函数 |

**2.2 检测项目**

源库：

|  |  |  |
| --- | --- | --- |
| 序号 | 检测内容 | 是否自测 |
| 1 | 源库分区表是否存在待迁移的分区 | 已自测 |
| 2 | 源库目标库字符集是否一致 | 已自测 |
| 3 | 源库表是否存在 | 已自测 |
| 4 | 源库表是否为分区表 | 已自测 |
| 5 | 源库分区表是否均为本地索引（目前该检测已注释） | 已自测 |
| 6 | 源库待迁移分区是否存在活动事务 | 已自测 |
| 7 | 源库分区表分区键内容是否可以转为合法日期（前端界面保存进也要检测） | 已自测 |
| 8 | 源库分区表中某一分区是否存在多月数据（前端界面保存进也要检测） | 已自测 |
| 9 | 源库分区表是否存在授权引用的用户（在imp函数中调用） | 已自测 |

目标库：

|  |  |  |
| --- | --- | --- |
| 序号 | 检测内容 | 是否自测 |
| 1 | 目标库中是否存在与源库相同的表空间 | 已自测 |
| 2 | 配置文件中记录的目标库用户是否存在 | 已自测 |
| 3 | 配置文件中记录的数据库链路是否存在 | 已自测 |
| 4 | 配置文件中记录的目录对象是否存在 | 已自测 |
| 5 | 配置文件中目录对象记录的路径是否有效 | 已自测 |
| 6 | 目标库分区中是否存在数据 | 已自测 |
| 7 | 目标库分区表索引状态是否有效 | 已自测 |

三、迁移前准备

（1）创建业务表空间

规则:与生产环境名称一样

示例:

|  |
| --- |
| create tablespace CLIC datafile size 16g;  alter tablespace clic add datafile size 16g;  alter tablespace clic add datafile size 16g;  create tablespace IDX\_CLIC datafile size 16g; |

（2）创业业务用户，授权

规则: 与生产环境名称一样

示例:

|  |
| --- |
| create user clic identified by "c43!cl76ic#)" default tablespace CLIC;  grant connect,resource,unlimited tablespace to CLIC;  grant create sequence to CLIC;  grant create synonym to CLIC;  grant create view to CLIC; |

（3）创建业务用户目录对象,需要在puppet用户下创建。

规则:用户名\_dump

示例:

|  |
| --- |
| conn puppet  create directory clic\_dump as '/logic\_dbbk/10.120.12.41/clic'; |

（4）创建业务用户数据库链路，需要在puppet用户下创建。

规则:dblink\_用户名

示例:

|  |
| --- |
| conn puppet  create database link dblink\_clic connect to puppet identified by "U2h7FliaezODyW18bbTo" using '10.120.12.41:1521/clic' ; |

（5）历史库建立历史库操作账号dbant专用账号

|  |
| --- |
| create user PUPPET identified by "XXXXXXXXXXXXXXX" default tablespace USERS temporary tablespace TEMP profile DEFAULT;  grant connect to PUPPET;  grant exp\_full\_database to PUPPET;  grant resource to PUPPET;  grant select\_catalog\_role to PUPPET;  grant create session to PUPPET;  grant execute any procedure to PUPPET;  grant select any dictionary to PUPPET;  grant unlimited tablespace to PUPPET;  grant CREATE DATABASE LINK to PUPPET;  grant create any directory to puppet;  grant imp\_full\_database to puppet; |

口令使用dbant项目中oracle数据库中puppet统一口令。

四、待完成功能

（1）前台页面开发

（2）脚本由python3.6改写为python2.7

（3）与晓阳写的程序集成。

（4）发短信，邮件功能没有增加。