分布式对象存储迁移工具介绍

# 使用说明

表1 工具说明

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 功能 |
| 1 | common\_upload.py | 上传指定目录下所有文件到MOS |
| 2 | common\_upload\_retry.py | 重新上传上次失败的文件到MOS |
| 3 | common\_upload\_check.py | 检查上传是否全部成功 |

表2 配置说明

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 内容说明 |
| 1 | config.py | 1. MOS相关配置   2、日志相关配置  3、指定数据库配置文件 |
| 2 | db.conf | 1、选择当前使用的数据库  2、配置当前使用的数据库信息 |

迁移工具全套包括3个脚本，如表1所示详细内容见下文介绍。

## 1.1全量上传

common\_upload.py

全量上传指定目录下的所有文件到MOS，该脚本会遍历config.py配置的DEFAULT\_UPLOAD\_DIR目录下所有文件，先去DB中检查该文件的cloudpath，判断该文件是否存在，如果不存在，则上传到MOS，并插入一条记录到DB；如果存在则跳过；如果文件上传MOS的过程中发生异常，上传失败，则记录一条失败日志，方便后续执行失败重传操作。

失败记录文件（区分数据库和s3便于排查）：

① upload\_files\_to\_MOS.log\_failed\_db

② upload\_files\_to\_MOS.log\_failed\_s3

## 1.2失败场景

以下是可能造成上传失败的场景

* 读取文件没有权限
* 分布式存储服务下线
* 数据源服务下线
* 文件名包含S3不支持的特殊字符
* 文件名包含DB不支持的特殊字符

## 1.3失败重传

common\_upload\_retry.py

给定上传失败的日志文件，从中顺序读取所有上传失败的文件路径，检查每个文件在DB中是否有过上传记录，若有则跳过，否则重新上传到MOS中，上传失败则记录失败记录日志，方便再次重传。

此步骤如果有失败，需要解决问题，重复执行，直到所有文件上传成功。

重传失败记录文件：

① upload\_files\_to\_MOS.log\_failed\_re-upload

## 1.4迁移验证

common\_upload\_check.py

遍历上传的目录所有文件，检查DB中是否有上传记录，如果有，则跳过；没有，则记录失败记录到日志文件。

若果执行完毕发现有失败记录，需要使用“失败重传”工具进行修补，重新上传。

检验记录文件：

1. 控制台打印; ②写入最新日志

# 注意事项

|  |  |
| --- | --- |
| 限制项 | 要求 |
| 运行操作系统 | Linux |
| Python要求 | Python2 |
| Oracle Lib库（若使用oracle） | cx\_Oracle预安装 |
| Mysql Lib 库 （若使用mysql） | MySQLdb预安装 |
| 磁盘空间 | 预留足够空间保存迁移日志，每200万文件需要1GB磁盘空间。 |
| 内存及CPU配置 | 尽量高 |