

数据迁移到 MaxCompute 的 N 种方式

想用阿里云大数据技术服务（MaxCompute），对于大多数人首先碰到的问题就是数据如何迁移到 MaxCompute 中。按照数据迁移场景，大致可以分为批量数据迁移和实时数据迁移两种，下面我们针对每种场景分别介绍几种常用方案。

一、异构数据源批量数据迁移到 MaxCompute

1、通过数加-数据开发（CDP）做数据同步

i. 开通数加开发环境，数据源需要配置到数加 DataIDE 中，并保证连通性。目前 MaxCompute 支持的数据源如下图：

新增数据源

* 数据源名称: 请输入数据源名称

数据源描述: 请输入数据源描述

* 数据源类型:

- ✓ rds
- mysql
- sqlserver
- postgresql
- odps
- ocs
- drds
- ads
- oss
- oracle
- ftp

 mysql

* RDS实例ID: 请输入RDS实例ID

* RDS实例购买者ID: 请输入RDS实例购买者ID, 请点击[这里](#)

* 数据库名: 请输入数据库名称

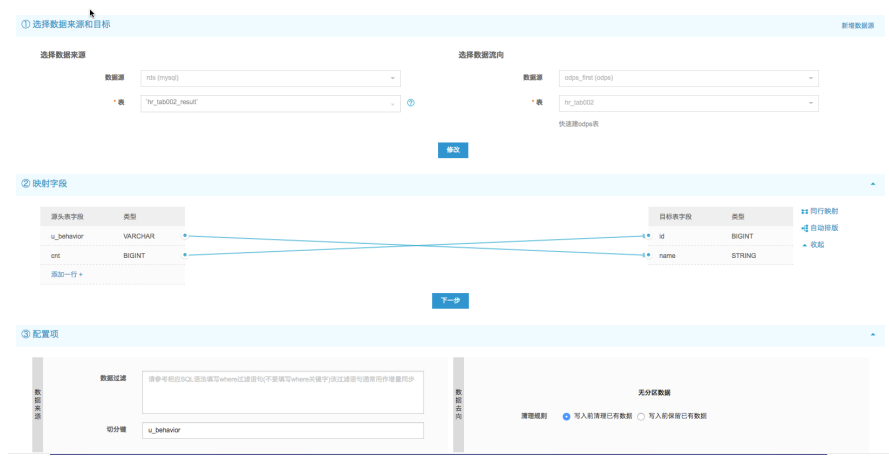
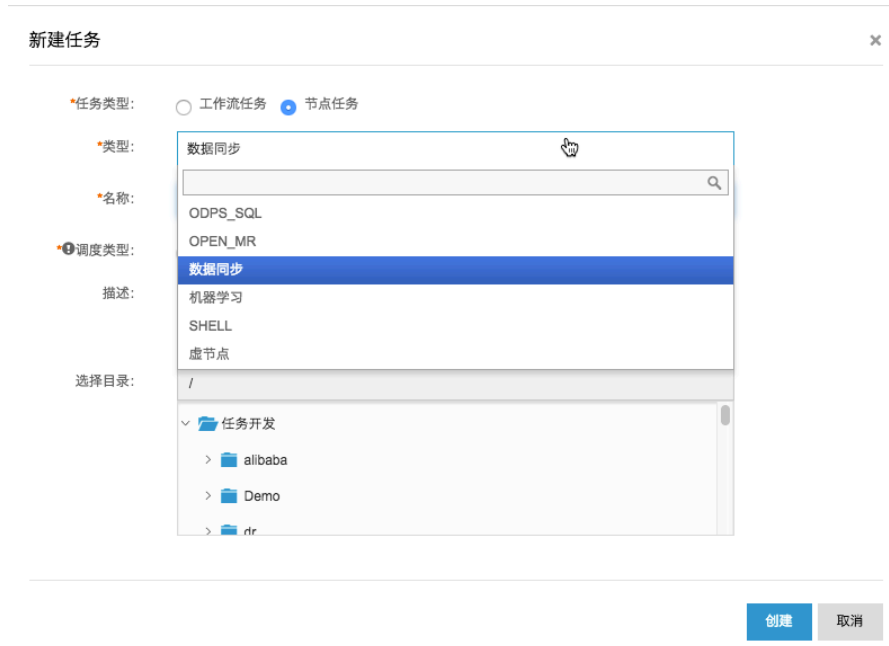
* 用户名: 请输入RDS用户名

* 密码: 请输入RDS密码

需要先添加RDS白名单才能连接成功哦, [点击查看如何添加白名单。](#)

测试连通性 确定 取消

ii. 创建数据同步任务，配置数据映射



iii. 保存后提交运行，可以通过执行日志监控执行成功与否。



使用限制：数加 DataIDE 中添加的数据源要确保在数据源类型支持列表中，并且要确保数据源连通性。

2、通过 DataX 实现数据同步

DataX 是阿里巴巴集团内被广泛使用的异构数据源离线同步工具，致力于实现包括关系型数据库(MySQL、Oracle 等)、HDFS、Hive、MaxCompute(原 ODPS)、HBase、FTP 等各种异构数据源之间稳定高效的数据同步功能。

DataX 本身作为离线数据同步框架，采用 Framework + plugin 架构构建。将数据源读取和写入抽象成为 Reader/Writer 插件，纳入到整个同步框架中。目前已经有了比较全面的插件体系，主流的 RDBMS 数据库、NOSQL、大数据计算系统都已经接入。DataX 目前支持数据如下：

类型	数据源	Reader(读)	Writer(写)
RDBMS 关系型数据库	MySQL	√	√
	Oracle	√	√
	SQL Server	√	√
	PostgreSQL	√	√
	达梦	√	√
	通用RDBMS(支持所有关系型数据库)	√	√
阿里云数仓数据存储	MaxCompute(原ODPS)	√	√
	Analytic DB(原ADS)		√
	OSS	√	√
	云数据库Memcache版(原OCS)	√	√
NoSQL数据存储	Table Store(原OTS)	√	√
	Hbase0.94	√	√
	Hbase1.1	√	√
	MongoDB	√	√
无结构化数据存储	TxtFile	√	√
	FTP	√	√
	HDFS	√	√

使用示例（从 MySQL 读取数据 写入 ODPS）：

- i. 直接下载 DataX 工具包，下载后解压至本地某个目录，修改权限为 755。下载地址：

<http://datax-opensource.oss-cn-hangzhou.aliyuncs.com/datax.tar.gz>

- ii. 创建作业配置文件

```
python datax.py -r mysqlreader -w odpswriter
```

- iii. 根据配置文件模板填写相关选项（源和目标数据库的用户名、密码、URL、表名、列名等），如下图：

```

{
  "job": {
    "content": [
      {
        "reader": {
          "name": "mysqlreader",
          "parameter": {
            "username": "****",
            "password": "****",
            "column": ["id", "age", "name"],
            "connection": [
              {
                "table": [
                  "test_table"
                ],
                "jdbcUrl": [
                  "jdbc:mysql://127.0.0.1:3306/test"
                ]
              }
            ]
          }
        },
        "writer": {
          "name": "odpswriter",
          "parameter": {
            "accessId": "****",
            "accessKey": "****",
            "column": ["id", "age", "name"],
            "odpsServer": "http://service.odps.aliyun.com/api",
            "partition": "pt='datax_test'",
            "project": "datax_opensource",
            "table": "datax_opensource_test",
            "truncate": true
          }
        }
      }
    ],
    "setting": {
      "speed": {
        "channel": 1
      }
    }
  }
}

```

iv. 启动 DataX 同步任务

python datax.py ./mysql2odps.json

3、通过 Sqoop 实现数据同步

请参考 <https://github.com/aliyun/aliyun-odps-sqoop>

4、通过 DTS（数据传输）实现数据同步

请参考 https://help.aliyun.com/document_detail/26612.html

二、本地文件上传到 MaxCompute

1、通过数加 DataIDE 导入本地文件

i. 登陆“数加-数据开发”，点击“导入-导入本地数据”



ii. 配置分隔符、数据文件字符编码等

本地数据导入

已选文件: data2.txt 只支持.txt、.csv和.log文件类型

分隔符号: 逗号 自定义

原始字符集: GBK

导入起始行: 1

首行为标题: 是

id	key
001	hello
002	world

下一步 取消

iii. 选择目标表后即可导入

本地数据导入

导入至表: jimi_test_tunnel 去新建表

字段匹配: 按位置匹配 按名称匹配

目标字段	源字段
id	id
name	空字段

上一步 导入 取消

使用限制：上传本地文件大小不能超过 10M。

2、通过 MaxCompute 客户端上传数据

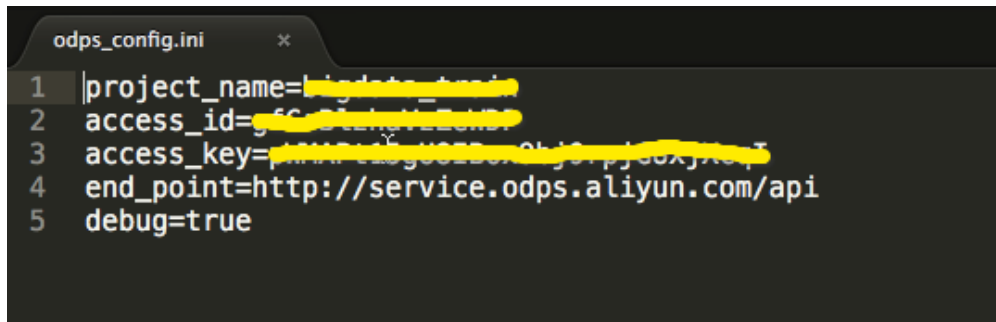
i. 下载 MaxCompute 客户端

下载路径：

http://repo.aliyun.com/download/odpscmd/0.24.1/odpscmd_public.zip

ii. 解压并配置客户端

解压后进入到 conf 目录，用编辑器打开 odps_config.ini，配置相应的 access_id、access_key、project_name 等。



```
odps_config.ini
1 project_name=bigdata_train
2 access_id=gfc-st-hq-1234567
3 access_key=MMF15G0220-013-pj0xjke7
4 end_point=http://service.odps.aliyun.com/api
5 debug=true
```

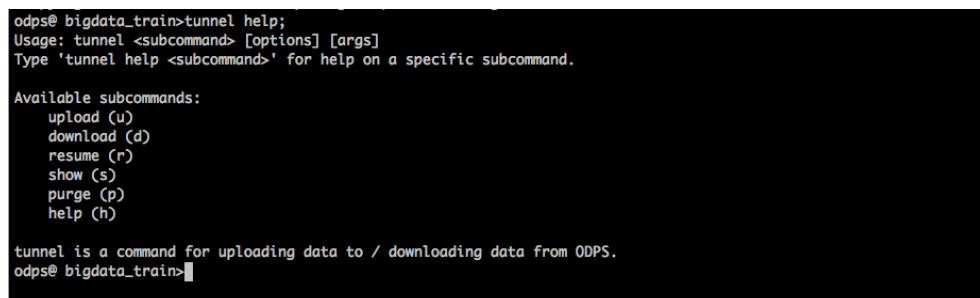
iii. 运行 MaxCompute 客户端

`odpscmd -config=../conf/odps_config.ini`



```
MacBook-Air:~ jimmy$ odpscmd --config=~/odpscmd_public/conf/odps_config.ini
[DEBUG]:ODPSConsole Start
Aliyun ODPS Command Line Tool
Version 0.24.1
©Copyright 2015 Alibaba Cloud Computing Co., Ltd. All rights reserved.
odps@ bigdata_train>
```

iv. 通过 tunnel 可以上传下载数据，详情可以通过 tunnel help 查看帮助



```
odps@ bigdata_train>tunnel help;
Usage: tunnel <subcommand> [options] [args]
Type 'tunnel help <subcommand>' for help on a specific subcommand.

Available subcommands:
  upload (u)
  download (d)
  resume (r)
  show (s)
  purge (p)
  help (h)

tunnel is a command for uploading data to / downloading data from ODPS.
odps@ bigdata_train>
```

v. 通过 tunnel upload 上传本地文件到 MaxCompute，详情可以通过 tunnel help upload 查看帮助

```

odps@ bigdata_train>tunnel help upload;
usage: tunnel 'upload [options] <path> <[project.]table[/partition]>'

        upload data from local file
-b, -block-size <ARG>      block size in MiB, default 100
-c, -charset <ARG>        specify file charset, default ignore.
                           set ignore to download raw data
-CP, -compress <ARG>      compress, default true
-dbr, -discard-bad-records <ARG> specify discard bad records
                           action(true|false), default false
-dfp, -date-format-pattern <ARG> specify date format pattern, default
                           yyyy-MM-dd HH:mm:ss
-fd, -field-delimiter <ARG> specify field delimiter, support
                           unicode, eg \u0001. default ","
-h, -header <ARG>        if local file should have table header,
                           default false
-mbr, -max-bad-records <ARG> max bad records, default 1000
-ni, -null-indicator <ARG> specify null indicator string, default
                           ""(empty string)
-rd, -record-delimiter <ARG> specify record delimiter, support
                           unicode, eg \u0001. default "\n"
-s, -scan <ARG>          specify scan file
                           action(true|false|only), default true
-sd, -session-dir <ARG>  set session dir, default
                           /Users/jimmy/Work/odpscmd_public/plugin
                           s/dship
-te, -tunnel_endpoint <ARG> tunnel endpoint
      -threads <ARG>      number of threads, default 1
-tz, -time-zone <ARG>   time zone, default local timezone:
                           Asia/Shanghai

Example:
tunnel upload log.txt test_project.test_table/p1="b1",p2="b2"
odps@ bigdata_train>

```

命令示例:

```
tunnel upload ./data.txt test_tunnel -fd "," -rd "\n";
```

解读:

data.txt – 数据文件

test_tunnel – MaxCompute 中数据表

-fd "," – 指定逗号为数据列分隔符

-rd "\n" – 指定换行符为数据行分隔符

备注: 通过 tunnel 上传数据比较灵活, 可以指定线程数等来提升效率。

另外有个性化需求的也可以通过 Tunnel SDK 的方式做数据同步, 详见:

https://help.aliyun.com/document_detail/27837.html

三、 实时数据归档到 MaxCompute

1. 通过 DataHub 将流式数据归档到 MaxCompute

用户通过创建 DataHub Connector，指定相关配置，即可创建将 Datahub 中流式数据定期归档的同步任务。请参考

<https://datahub.console.aliyun.com/intro/advancedguide/connector.html>

2. 通过 DTS 将数据实时同步到 MaxCompute

请参考 https://help.aliyun.com/document_detail/26614.html

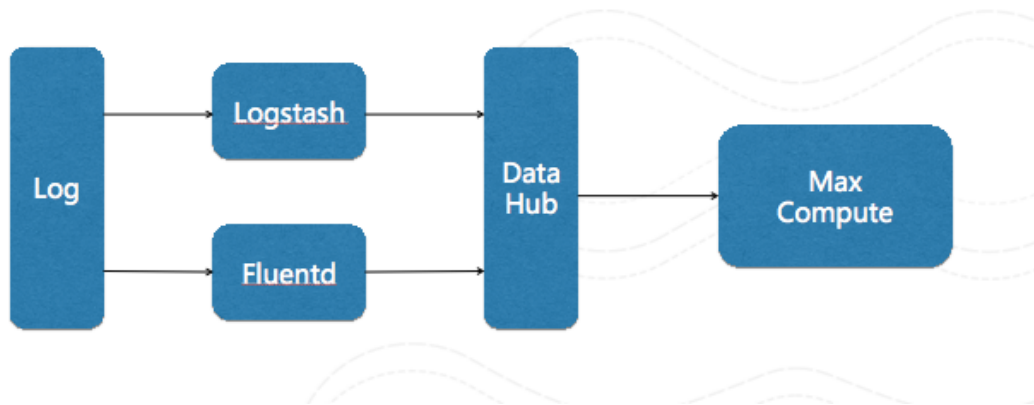
3. 通过 OGG 将数据实时同步到 MaxCompute

这种方式要通过 OGG 将实时数据先同步到 DataHub，再在 DataHub 中通过创建 DataHub Connector 将数据实时归档到 MaxCompute。请参考

<https://datahub.console.aliyun.com/intro/guide/plugins/ogg.html>

四、日志数据同步到 MaxCompute

目前日志类型的数据实时同步到 MaxCompute 的需求也非常强。市面上也有很多成熟的日志收集工具，比如 Fluentd、Logstash。日志数据实时同步到 MaxCompute 的方案也是要借助于这些成熟的日志收集工具，将日志数据同步到 DataHub 中后，再通过 DataHub 将数据归档到 MaxCompute，数据链路：



1. 通过 Logstash 采集日志数据到 MaxCompute

请参考

<https://datahub.console.aliyun.com/intro/guide/plugins/logstash.html>

2. 通过 Fluentd 采集日志数据到 MaxCompute

请参考

<https://datahub.console.aliyun.com/intro/guide/plugins/fluentd.html>