**数据迁移 MySQL 2 OB**

**环境准备**

**准备 MySQL 环境**

# Docker 下部署 Mysql 5.7

docker run -itd --name mysql -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=root mysql:5.7

docker exec -it mysql bash

**准备测试数据**

# 登入MySQL

mysql -u root -p root

-- 建库

create database test\_data;

use test\_data;

--建表

CREATE TABLE `Student`(`s\_id` VARCHAR(20),`s\_name` VARCHAR(20) NOT NULL DEFAULT '',`s\_birth` VARCHAR(20) NOT NULL DEFAULT '',`s\_sex` VARCHAR(10) NOT NULL DEFAULT '',PRIMARY KEY(`s\_id`));

CREATE TABLE `Course`(`c\_id` VARCHAR(20),`c\_name` VARCHAR(20) NOT NULL DEFAULT '',`t\_id` VARCHAR(20) NOT NULL,PRIMARY KEY(`c\_id`));

CREATE TABLE `Teacher`(`t\_id` VARCHAR(20),`t\_name` VARCHAR(20) NOT NULL DEFAULT '',PRIMARY KEY(`t\_id`));

CREATE TABLE `Score`(`s\_id` VARCHAR(20),`c\_id` VARCHAR(20),`s\_score` INT(3),PRIMARY KEY(`s\_id`,`c\_id`));

--入数

insert into Student values('01' , '赵雷' , '1990-01-01' , '男');

insert into Student values('02' , '钱电' , '1990-12-21' , '男');

insert into Student values('03' , '孙风' , '1990-05-20' , '男');

insert into Student values('04' , '李云' , '1990-08-06' , '男');

insert into Student values('05' , '周梅' , '1991-12-01' , '女');

insert into Student values('06' , '吴兰' , '1992-03-01' , '女');

insert into Student values('07' , '郑竹' , '1989-07-01' , '女');

insert into Student values('08' , '王菊' , '1990-01-20' , '女');

insert into Course values('01' , '语文' , '02');

insert into Course values('02' , '数学' , '01');

insert into Course values('03' , '英语' , '03');

insert into Teacher values('01' , '张三');

insert into Teacher values('02' , '李四');

insert into Teacher values('03' , '王五');

insert into Score values('01' , '01' , 80);

insert into Score values('01' , '02' , 90);

insert into Score values('01' , '03' , 99);

insert into Score values('02' , '01' , 70);

insert into Score values('02' , '02' , 60);

insert into Score values('02' , '03' , 80);

insert into Score values('03' , '01' , 80);

insert into Score values('03' , '02' , 80);

insert into Score values('03' , '03' , 80);

insert into Score values('04' , '01' , 50);

insert into Score values('04' , '02' , 30);

insert into Score values('04' , '03' , 20);

insert into Score values('05' , '01' , 76);

insert into Score values('05' , '02' , 87);

insert into Score values('06' , '01' , 31);

insert into Score values('06' , '03' , 34);

insert into Score values('07' , '02' , 89);

insert into Score values('07' , '03' , 98);

输出

 Query OK, 0 rows affected (0.01 sec)

 Query OK, 0 rows affected (0.01 sec)

 Query OK, 0 rows affected (0.01 sec)

 Query OK, 0 rows affected (0.01 sec)

 Query OK, 1 row affected (0.00 sec)

 ......

 Query OK, 1 row affected (0.00 sec)

 Query OK, 1 row affected (0.00 sec)

 Query OK, 1 row affected (0.00 sec)

 Query OK, 1 row affected (0.00 sec)

**安装 datax**

安装到 Mysql 所属容器

# 下载 解压

cd /tmp && wget http://datax-opensource.oss-cn-hangzhou.aliyuncs.com/datax.tar.gz

tar -zxf datax.tar.gz -C .

# 删除异常的文件

rm -rf /tmp/datax/plugin/.\_\*

rm -rf /tmp/datax/plugin/\*/.\_\*

# 安装验证

cd /tmp/datax/bin && python datax.py ../job/job.json

输出

2022-02-17 18:06:41.960 [job-0] INFO JobContainer -

任务启动时刻 : 2022-02-17 18:06:31

任务结束时刻 : 2022-02-17 18:06:41

任务总计耗时 : 10s

任务平均流量 : 253.91KB/s

记录写入速度 : 10000rec/s

读出记录总数 : 100000

读写失败总数 : 0

**准备 OB 环境**

docker run -itd -p 2881:2881 -p 2883:2883 -m 10G --name obce obpilot/oceanbase-ce

docker exec -it obce bash

obd cluster start obdemo

obclient -h 127.1 -uroot@obmysql -P2881 -p123456 -c -A test

**连接 OB 与 MySQL 所在容器**

docker network create data\_trans

docker network connect data\_trans obce

docker network connect data\_trans mysql

**基于 mysqldump 的数据迁移**

**导出MySQL数据**

* -d 仅导出表结构
* -t 仅导出表数据
* > file.sql  直接输出到指定的文件
* --compact 压缩输出

mysqldump -h 127.1 -uroot -P3306 -proot test\_data --compact > mysql\_export\_all.sql

节选输出

-- cat mysql\_export\_all.sql

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `Course` (

 `c\_id` varchar(20) NOT NULL,

 `c\_name` varchar(20) NOT NULL DEFAULT '',

 `t\_id` varchar(20) NOT NULL,

 PRIMARY KEY (`c\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

INSERT INTO `Course` VALUES ('01','è¯­æ–‡','02'),('02','æ•°å­¦','01'),('03','è‹±è¯­','03');

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `Score` (

 `s\_id` varchar(20) NOT NULL,

 `c\_id` varchar(20) NOT NULL,

 `s\_score` int(3) DEFAULT NULL,

 PRIMARY KEY (`s\_id`,`c\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

INSERT INTO `Score` VALUES ('01','01',80),('01','02',90),('01','03',99),('02','01',70),('02','02',60),('02','03',80),('03','01',80),('03','02',80),('03','03',80),('04','01',50),('04','02',30),('04','03',20),('05','01',76),('05','02',87),('06','01',31),('06','03',34),('07','02',89),('07','03',98);

...

**标准化导出文件**

* mysqldump 会导出注释后的视图DML,不需要时可以删除
* mysqldump 会涵盖部分MySQL特有的语法,需要移除后才能正确导入OB

# 替换 OB 不支持参数

sed -i 's/MAX\_ROWS=/; -- MAX\_ROWS=/' mysql\_export\_ddl.sql

sed -i 's/CHARACTER\ SET\ latin1//g' mysql\_export\_ddl.sql

**导入数据到 OB**

**上传文件到OB集群所在容器**

 docker cp /tmp/mysql\_export\_all.sql obce:/tmp/

**登录OB集群**

obclient -h 127.1 -uroot@obmysql -P2881 -c -A oceanbase

**数据导入前检查**

* DML中包含外键时需要先暂时禁用外键约束

set global foreign\_key\_checks=off; -- 禁用外键约束

show global variables like '%foreign%'; -- 检查配置是否生效 ,OFF 代表禁用

* DML中表名大写时,建议检查源端是否配置大小写敏感,OB需要与之保持一致

**导入数据**

**注意** 双横线的注释在导入OB时会报错

source /tmp/mysql\_export\_all.sql

输出

MySQL [test\_data]> source /tmp/mysql\_export\_all.sql

Query OK, 1 row affected (0.011 sec)

Database changed

Query OK, 0 rows affected (0.031 sec)

......

Query OK, 1 row affected (0.001 sec)

MySQL [test\_data]> show databases;

+--------------------+

| Database |

+--------------------+

| oceanbase |

| information\_schema |

| mysql |

| test |

| test\_data |

+--------------------+

5 rows in set (0.002 sec)

MySQL [test\_data]> use test\_data;

Database changed

MySQL [test\_data]> show tables;

+---------------------+

| Tables\_in\_test\_data |

+---------------------+

| course |

| score |

| student |

| teacher |

+---------------------+

4 rows in set (0.002 sec)

MySQL [test\_data]> select count(\*) from score;

+----------+

| count(\*) |

+----------+

| 18 |

+----------+

1 row in set (0.001 sec)

**基于 datax 的数据迁移**

MySQL 数据离线同步到 OceanBase租户

**编写配置文件**

vim /tmp/datax/job/mysql2ob.json

{

 "job": {

 "setting": {

 "speed": {

 "channel": 4

 },

 "errorLimit": {

 "record": 0,

 "percentage": 0.1

 }

 },

 "content": [

 {

 "reader": {

 "name": "mysqlreader",

 "parameter": {

 "username": "root",

 "password": "root",

 "column": [

 "\*"

 ],

 "connection": [

 {

 "table": [

 "Score"

 ],

 "jdbcUrl": ["jdbc:mysql://mysql:3306/test\_data?useUnicode=true&characterEncoding=utf8&useSSL=false"]

 }

 ]

 }

 },

 "writer": {

 "name": "oceanbasev10writer",

 "parameter": {

 "obWriteMode": "insert",

 "column": [

 "\*"

 ],

 "preSql": [

 "truncate table Score"

 ],

 "connection": [

 {

 "jdbcUrl":"||\_dsc\_ob10\_dsc\_||obce-single:obmysql||\_dsc\_ob10\_dsc\_||jdbc:oceanbase://obce:2883/test\_data?useLocalSessionState=true&allowBatch=true&allowMultiQueries=true&rewriteBatchedStatements=true",

 "table": [

 "Score"

 ]

 }

 ],

 "username": "root",

 "password":"123456",

 "writerThreadCount":10,

 "batchSize": 1000,

 "memstoreThreshold": "0.9"

 }

 }

 }

 ]

 }

}

**迁移前需要提前在OB中定义好库和表**

-- mysql -h obce -uroot@obmysql#obce-single -P2883 -p -c -A test 登录 OB

create database test\_data;

use test\_data;

CREATE TABLE `Score` ( `s\_id` varchar(20) NOT NULL, `c\_id` varchar(20) NOT NULL, `s\_score` int(3) DEFAULT NULL, PRIMARY KEY (`s\_id`,`c\_id`) );

mysql> show tables;

+---------------------+

| Tables\_in\_test\_data |

+---------------------+

| score |

+---------------------+

1 row in set (0.01 sec)

mysql> select count(\*) from score;

+----------+

| count(\*) |

+----------+

| 0 |

+----------+

1 row in set (0.00 sec)

**执行数据迁移任务**

cd /tmp/datax/bin && python datax.py ../job/mysql2ob.json

输出

2022-02-17 19:08:14.632 [job-0] INFO JobContainer -

任务启动时刻 : 2022-02-17 19:08:04

任务结束时刻 : 2022-02-17 19:08:14

任务总计耗时 : 10s

任务平均流量 : 10B/s

记录写入速度 : 1rec/s

读出记录总数 : 18

读写失败总数 : 0

核验结果

mysql> select count(\*) from score;

+----------+

| count(\*) |

+----------+

| 18 |

+----------+