

1. MySQL 同步增量数据到 OceanBase

1.1 简介

简单模拟下 MySQL 到 OB 数据同步的过程，包括 schema 同步及 单个表 同步过程，仅记录操作过程。

1.2 环境信息

1.2.1 OB集群

3台 centos (docker 环境)部署的OB集群，端口 2882:2881 ,服务器 IP 分别为 172.20.0.10、172.20.0.11、172.20.0.12。

本次未使用 obproxy。

```
1 # 在集群上创建一个租户 obcp_t1 ，过程略。
2 # 建立用户 obcanal
3 create user obcanal@'%' identified by 'obcanal' ;
4 GRANT ALL PRIVILEGES ON *.* TO 'obcanal' ;
5
6 create database mydb1 ; # 用户测试整库同步
7 create database mydb2 ; # 用于测试单表同步
```

1.2.2 MySQL 服务器

1台 MySQL 5.7, IP 地址为 172.20.0.32, 端口3306。

```
1 docker run --name mysql57 -p 33306:3306 -e MYSQL_ROOT_PASSWORD=password -e TZ=Asia/Shanghai --restart=always -v /data/dockerdata/mysql1/conf:/etc/mysql -v /data/dockerdata/mysql1/datadir:/var/lib/mysql --network=adnc_net --ip 172.20.0.32 -d mysql:5.7
```

配置 MySQL

```
1 # 在/data/dockerdata/mysql1/conf/my.cnf文件中加入内容 (3个参数为必改项)
2 [mysqld]
3 log-bin=mysql-bin # 开启 binlog
4 binlog-format=ROW # 选择 ROW 模式
5 server_id=1 # 配置 MySQL replaction 需要定义
6 skip_ssl # 禁用ssl, 否则jdbc无法连接
```

创建 canal 连接 MySQL 账户

```
1 # 授权 canal 链接 MySQL 账号具有作为 MySQL slave 的权限, 如果已有账户可直接 grant
2 CREATE USER canal@'%' IDENTIFIED BY 'canal';
3 GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'canal'@'%';
4 FLUSH PRIVILEGES;
```

创建测试库

```
1 create database mydb1;
```

1.2.3 canal、adapter 服务器

1台 centos (docker 环境), IP 地址为 172.20.0.31, 用于部署 canal、canal-adapter

```
1 # `canal` 运行需要 `java` 环境, 参考安装命令:
2 sudo yum -y install java-1.8.0-openjdk.x86_64
```

1.2.4 附 docker 快速构建服务器说明

```
1  docker network create --driver bridge --subnet=172.20.0.0/16 --  
    gateway=172.20.0.1 adnc_net # 自定义网络  
2  docker run --privileged -itd -p 2230:22 --name c10 --network adnc_net --ip  
    172.20.0.10 centos:7 /usr/sbin/init # 运行容器并指定IP  
3  docker exec -it c10 bash # 进入容器  
4  # 更新centos并安装必要的软件包  
5  yum check-update -y && yum update -y  
6  yum install initscripts screen wget -y && yum install passwd -y && yum  
    install net-tools -y && yum install -y unzip zip && yum -y install wget tree  
    && yum install -y sudo net-tools  
7  yum install -y openssh-server openssh-clients  
8  systemctl start sshd  
9  useradd admin # 添加admin用户  
10 passwd admin # 修改密码
```

1.3 安装canal

从 <https://github.com/alibaba/canal/releases> 下载的包如

下: canal.deployer-1.1.5.tar.gz、 canal.adapter-1.1.5.tar.gz

登录 canal、 adapter 服务器（172.20.0.31）,并将上面的两个软件上传到该服务器。

```
1  # 以下命令均适用root进行。  
2  mkdir ~/canal  
3  tar zxvf canal.deployer-1.1.5.tar.gz -C ~/canal
```

1.3.1 修改配置

```
1 # 修改/root/canal/conf/canal.properties文件，（为简洁，去除部分内容）：
2 more /root/canal/conf/canal.properties
3 # tcp, kafka, rocketMQ, rabbitMQ
4 canal.serverMode = tcp # 指定该值为tcp，默认值即为tcp。
```

```
1 # 修改/root/canal/conf/example/instance.properties文件，修改后的内容如下（为简洁，去
  除部分内容）：
2 more /root/canal/conf/example/instance.properties
3 # position info
4 canal.instance.master.address=172.20.0.32:3306 ## 指定MySQL服务器信息
5 .....
6 # username/password
7 canal.instance.dbUsername=canal ## 指定前面部分创建的MySQL账户名
8 canal.instance.dbPassword=canal ## 指定前面部分创建的MySQL账户密码
9
```

1.3.2 启动

```
1 /root/canal/bin/startup.sh
2 # 如有如下报错，参考前面部分安装java环境。
3 bin/startup.sh: line 30: which: command not found # 这里缺少 which命令，可以 yum
  install -y which 安装
4 Cannot find a Java JDK. Please set either set JAVA or put java (>=1.5) in your
  PATH.
```

1.3.3 查看日志

1.3.3.1 查看 server 日志

```
1 tail -500f /root/canal/logs/canal/canal.log</pre>  
2 # 有 [main] INFO com.alibaba.otter.canal.deployer.CanalStarter - ## the canal  
server is running now ..... 表示启动成功。
```

1.3.3.2 查看 instance 的日志

```
1 tail -500f /root/canal/logs/example/example.log  
2 # 确保没有ERROR, 并有如下信息显示 (源端MySQL binlog信息)  
3 # 2021-12-15 17:25:39.346 [destination = example , address = /172.20.0.32:3306 ,  
EventParser] WARN c.a.o.c.p.inbound.mysql.rds.RdsBinlogEventParserProxy - --->  
find start position successfully,  
EntryPosition[included=false,journalName=replicas-mysql-  
bin.000005,position=1578,serverId=100,gtid=,timestamp=1639556730000] cost :  
1332ms , the next step is binlog dump
```

1.3.4 关闭命令

```
1 sh bin/stop.sh
```

1.4 部署 RDB 适配器

RDB adapter 用于适配 **MySQL** 到任意关系型数据库(需支持 **jdbc**)的数据同步及导入。

1.4.1 安装 canal-adapter

```
1 # 以下命令均适用root进行。  
2 mkdir ~/canal-adapter  
3 tar zvxvf canal.adapter-1.1.5.tar.gz -C ~/canal-adapter/
```

1.4.2 修改配置

```
1 # more ~/canal-adapter/conf/application.yml
2
3 srcDataSources:
4   defaultDS:
5     url: jdbc:mysql://172.20.0.32:3306/mydb1?useUnicode=true
6     username: canal
7     password: canal
8   canalAdapters:
9     - instance: example # canal instance Name or mq topic name
10     groups:
11       - groupId: g1
12         outerAdapters:
13           - name: logger
14           - name: rdb
15             key: mysql1
16             properties:
17               jdbc.driverClassName: com.mysql.jdbc.Driver
18               jdbc.url: jdbc:mysql://172.20.0.10:2881/mydb1
19               jdbc.username: obcanal@obcp_t1
20               jdbc.password: obcanal
21           - name: rdb
22             key: mysql2
23             properties:
24               jdbc.driverClassName: com.mysql.jdbc.Driver
25               jdbc.url: jdbc:mysql://172.20.0.10:2881/mydb2
26               jdbc.username: obcanal@obcp_t1
27               jdbc.password: obcanal
28
```

1.4.3 增加 RDB 文件

```
1 # 整库同步
2 more ~/canal-adapter/conf/rdb/mydb1.yml
3 ## Mirror schema synchronize config
4 dataSourceKey: defaultDS
5 destination: example
6 groupId: g1
7 outerAdapterKey: mysql1
8 concurrent: true
9 dbMapping:
10   mirrorDb: true
11   database: mydb1 # 库名和源库一样
12
13 # 单表同步
14 more ~/canal-adapter/conf/rdb/mydb2.yml
15
16 dataSourceKey: defaultDS
17 destination: example
18 groupId: g1
19 outerAdapterKey: mysql2
20 concurrent: true
21 dbMapping:
22   database: mydb1 # 源库名
23   table: t1 # 源表名
24   targetTable: t2 # 目标端表名, 库名在jdbc配置里。
25   targetPk:
26     id: id
27   mapAll: true
28   commitBatch: 3000
29
30
```

1.4.4 启动

```
1 ~/canal-adapter/bin/startup.sh
```

1.4.5 查看日志

```
1 tail -500f ~/canal-adapter/logs/adapter/adapter.log # 确保无错误发生
```

1.5 测试

过程省略

1.6 问题记录

1.6.1 ssl 禁用

第一种方法：在源端 `jdbc` 连接串,增加 `&useSSL=false`

第二种方法：在 `MySQL` 配置文件中加入：`skip_ssl`，重启 `MySQL` 生效

1.6.2 DML不生效，DDL生效

`MySQL` 的 `binlog` 格式设置为 `ROW`。

1.6.3 单表同步

需要在目标端先建表，整个库同步不需要单独建表。

1.6.4 表要主键或唯一键

要同步的表需要有主键或唯一键