**测试前的准备**

首先要安装obclient，安装obclient后会自动带一个mysqltest文件。

yum install obclient

OB的很多单元测试会直接使用指定用户名连接指定租户，所以需要提前创建好相关的租户和用户。

创建租户mysql

obd cluster tenant create unittest --tenant-name=mysql # 我这里部署的cluster 名字叫unittest

sys租户的root用户，没有密码

创建用户 admin@mysql 密码admin

创建用户 admin@sys 密码 admin

创建用户的方法：

obclient -uroot -h 127.0.0.1 -P 2881 #不指定租户，默认连接sys租户，就会创建sys租户下的用户

obclient -uroot@mysql -h 127.0.0.1 -P 2881 # 使用-uroot@mysql ，连接mysql租户，这时创建的用户就是mysql租户下的

连接后，执行下面的命令创建用户

CREATE USER 'admin' IDENTIFIED BY 'admin';

GRANT ALL PRIVILEGES ON \*.\* TO admin WITH GRANT OPTION;

测试是否可以连接成功：

obclient -uadmin@mysql -h 127.0.0.1 -P 2881 -p

如果连接不成功，可以执行下这个命令，再测试：

ALTER TENANT mysql SET VARIABLES ob\_tcp\_invited\_nodes='%';

执行单元测试有两种方法。

**使用obd做单元测试**

NOTE: 可以参考源码中的说明文件 test/README.md

在mysql\_test的目录下

obd test mysqltest `cluster-name` --all # 测试所有用例

obd test mysqltest `cluster-name` --suite=join # 测试join下的所有用例

OBD在启动每一个测试用例前，都会执行下面的命令：

obclient -h127.0.0.1 -P2881 -uroot -A -Doceanbase -e "alter system set \_enable\_static\_typing\_engine = True;select sleep(2);"

obd会按照指定参数，遍历所有case，对每个case，会执行下面的命令运行单元测试：

/u01/obclient/bin/mysqltest --host=127.0.0.1 --port=2881 --tmpdir=$TEST\_TMP\_BASE\_DIR/tmp --logdir=$TEST\_TMP\_BASE\_DIR/var/log --silent --user=admin@mysql --database=test --timer-file=$TEST\_TMP\_BASE\_DIR/var/log/timer --tail-lines=20 --test-file=./mysql\_test/test\_suite/t/${case}.test --result-file=./mysql\_test/test\_suite/r/mysql/${case}.result

其中 TEST\_TMP\_BASE\_DIR 是一个临时目录，可以是/tmp。

/u01/obclient/bin/mysqltest 是安装obclient后，mysqltest的路径。

obd默认会在当前目录下，查找mysql\_test/test\_suite下的文件。

开源版本的OB，只有MySQL模式，所有的result文件，都在每个test目录对应的result目录——r——下的mysql目录，比如，mysql\_test/test\_suite/t/join/t下保存了所有的test文件，mysql\_test/test\_suite/r/mysql下保存了所有的result文件。

**运行脚本执行进行测试**

了解obd的运行原理后，就可以自己写脚本进行测试了。

直接上脚本吧

NOTE：这个脚本在群文件中给出，需要注意的是，这个脚本需要在mysql\_test同一层级运行，比如压缩包在HOME解压，那么需要在HOME运行 ./mysql\_test/mysql-test.sh -s join

#!/bin/bash

MYSQL\_TEST\_BIN=/u01/obclient/bin/mysqltest

TEST\_BASE\_DIR=$PWD

# 有些test文件会引用mysql\_test/include目录下的头文件，所以这个脚本需要在mysql\_test的上层执行

TEST\_SUITE\_DIR=$TEST\_BASE\_DIR/mysql\_test/test\_suite

#RECORD\_RESULT="-r"

RECORD\_RESULT=

# OB的测试用例中，需要这几个环境变量

OBMYSQL\_MS0=127.0.0.1

OBMYSQL\_PORT=2881

OBMYSQL\_USR=admin

OBMYSQL\_PWD=admin

OBMYSQL\_DB=test

SUITES=

export OBMYSQL\_MS0

export OBMYSQL\_PORT

export OBMYSQL\_USR

export OBMYSQL\_PWD

export OBMYSQL\_DB

while getopts 's:' OPT;do

case $OPT in

s)

SUITES="$OPTARG";;

?)

echo "Usage `basename $0` [-s suite\_name]"

esac

done

if [ "x${SUITES}" == "x" ];

then

SUITES=`ls $TEST\_SUITE\_DIR`

fi

# 开源分支代码，现在测试都开启static\_engine功能

obclient -h127.0.0.1 -P2881 -uroot -A -Doceanbase -e 'alter system set \_enable\_static\_typing\_engine = True;select sleep(2);'

for suite in $SUITES

do

suite\_dir=${TEST\_SUITE\_DIR}/${suite}

for testname in `ls ${suite\_dir}/t/\*.test`

do

case=`basename $testname .test`

echo test ${suite}.${case}

begin=$[$(date +%s%N)]

$MYSQL\_TEST\_BIN --host=127.0.0.1 --port=2881 --tmpdir=$TEST\_BASE\_DIR/tmp --logdir=$TEST\_BASE\_DIR/var/log --silent --user=$OBMYSQL\_USR@mysql --password=$OBMYSQL\_PWD --database=$OBMYSQL\_DB --timer-file=$TEST\_BASE\_DIR/var/log/timer --tail-lines=20 --max-connect-retries=3 --connect-timeout=10 --test-file=${suite\_dir}/t/${case}.test --result-file=${suite\_dir}/r/mysql/${case}.result $RECORD\_RESULT

result\_code=$?

end=$[$(date +%s%N)]

result\_string='success'

if [ $result\_code -ne 0 ]

then

result\_string='failed'

fi

echo test ${suite}.${case} done $result\_string . cost $[(end-begin)/1000000] ms

if [ $result\_code -ne 0 ]

then

echo "Bye"

exit 1 # 一旦有错误，就直接退出，不再继续测试

fi

done

done