## 实践练习六（必选）：查看 OceanBase 执行计划

本次练习是必选练习之一。

练习目的

本次练习目的掌握 OceanBase 的执行计划查看方法，包括 explain 命令和查看实际执行计划。

练习条件

有服务器，内存资源至少 12G\*1 台，部署有 OceanBase 集群（单副本或三副本都可以）。

练习内容

请记录并分享下列内容：

（必选）1、使用 BenmarkSQL 运行 TPC-C ，并发数不用很高，5~10 并发即可（根据机器资源）。

（必选）2、分析 TPC-C TOP SQL，并查看 3条 SQL 的 解析执行计划 和 实际执行计划。

**一、服务器信息**

|  |  |
| --- | --- |
| IP地址 | 10.0.14.22 |
| 网卡名 | enp0s8 |
| OS | CentOS Linux release 7.8.2003 (Core) |
| CPU | 16C |
| 内存 | 20G |
| 磁盘1 | 本地盘 /dev/sdd (lvm /data 、 /redo) |

二、参数设置

2.1、OS 参数

sh -c 'for x in /sys/class/net/enp0s8/queues/rx-\*; do echo ff>$x/rps\_cpus; done'

sh -c "echo 32768 > /proc/sys/net/core/rps\_sock\_flow\_entries"

sh -c "echo 4096 > /sys/class/net/eth0/queues/rx-0/rps\_flow\_cnt"

#### 2.2、OceanBase 参数

##### 2.2.1 SYS 租户参数

obclient -h 10.0.14.22 -P 2883 -uroot@sys#obcluster -p -c -A oceanbase

alter system set enable\_auto\_leader\_switch=false;

alter system set enable\_one\_phase\_commit=false;

alter system set enable\_monotonic\_weak\_read = true;

alter system set weak\_read\_version\_refresh\_interval='5s';

alter system \_ob\_minor\_merge\_schedule\_interval='5s';

alter system set memory\_limit\_percentage = 90;

alter system set memstore\_limit\_percentage = 55;

alter system set freeze\_trigger\_percentage = 70;

alter system set minor\_freeze\_times = 50;

alter system set minor\_warm\_up\_duration\_time = 0;

alter system set merge\_thread\_count = 32;

alter system set minor\_merge\_concurrency = 8;

alter system set \_mini\_merge\_concurrency = 4;

##### 2.2.2 PROXY 参数

alter proxyconfig set proxy\_mem\_limited='4G'; --防止 oom，可根据实际环境动态调整

alter proxyconfig set enable\_compression\_protocol=false; --关闭压缩，降低 CPU 百分率

alter proxyconfig set work\_thread\_num=32; -- 调整工作线程数，寻找最优性能

alter proxyconfig set enable\_compression\_protocol=false;

alter proxyconfig set enable\_metadb\_used=false;

alter proxyconfig set enable\_standby=false;

alter proxyconfig set enable\_strict\_stat\_time=false;

alter proxyconfig set use\_local\_dbconfig=true;

create database tpcc;

unzip benchmarksql-5.0-master.zip

##### 3.2 配置

编辑 /root/benchmarksql/run/props.ob 目录下,编辑后的内容如下:

db=mysql

driver=com.alipay.oceanbase.obproxy.mysql.jdbc.Driver

conn=jdbc:oceanbase://127.0.0.1:2883/tpcc?useUnicode=true&characterEncoding=utf-8user=root@sys#obcluster password=123456

warehouses=2

loadWorkers=10

terminals=10

//To run specified transactions per terminal- runMins must equal zero

runTxnsPerTerminal=0

//To run for specified minutes- runTxnsPerTerminal must equal zero

runMins=1

//Number of total transactions per minute

limitTxnsPerMin=0

//Set to true to run in 4.x compatible mode. Set to false to use the

//entire configured database evenly.

terminalWarehouseFixed=true

//The following five values must add up to 100

newOrderWeight=45

paymentWeight=43

orderStatusWeight=4

deliveryWeight=4

stockLevelWeight=4

// Directory name to create for collecting detailed result data.

// Comment this out to suppress.

resultDirectory=my\_result\_%tY-%tm-%td\_%tH%tM%tS

osCollectorScript=./misc/os\_collector\_linux.py

osCollectorInterval=1

##### 3.3 运行创建表语句

sh runSQL.sh props.ob sql.common/tableCreates.sql



##### 四、测试

##### 4.1 加载数据

sh runLoader.sh props.ob





##### 4.2 创建索引

##### ##登陆到数据库,创建两个索引(表上之前是没有索引的)

[root@ob02 run]# obclient -h 10.0.14.22 -uroot@sys#obcluster -P2883 -p'123456' -c -A tpcc

create index bmsql\_customer\_idx1 on bmsql\_customer (c\_w\_id, c\_d\_id, c\_last, c\_first) local;

create index bmsql\_oorder\_idx1 on bmsql\_oorder (o\_w\_id, o\_d\_id, o\_carrier\_id, o\_id) local;

##### **4.3 执行性能测试**

sh runBenchmark.sh props.ob



#### 五、TPC-C TOP SQL分析

#### 5.1 查询TOP10 sql

SELECT sql\_id, count(\*), round(avg(elapsed\_time)) avg\_elapsed\_time, round(avg(execute\_time)) avg\_exec\_time

       FROM gv$sql\_audit s

       WHERE 1=1

        and user\_name='tpcc'

        and request\_time >= time\_to\_usec(DATE\_SUB(current\_timestamp, INTERVAL 30 MINUTE) )

       GROUP BY sql\_id

       order by avg\_elapsed\_time desc limit 10;



##### 5.2 对 elapsed 时间最长的前三条 sql 进行分析

SELECT sql\_id, count(\*), round(avg(elapsed\_time)) avg\_elapsed\_time,

                     round(avg(execute\_time)) avg\_exec\_time,

                     s.svr\_ip,

                     s.svr\_port,

                     s.tenant\_id,

                     s.plan\_id

                    FROM gv$sql\_audit s

                    WHERE 1=1

                     and request\_time >= time\_to\_usec(DATE\_SUB(current\_timestamp, INTERVAL 30 MINUTE) )

                    GROUP BY sql\_id

                   order by avg\_elapsed\_time desc limit 3;



select distinct query\_sql from gv$sql\_audit where sql\_id='5C9973474F2AF4CB0C20B54B9B49A2D7';



select distinct query\_sql from gv$sql\_audit where sql\_id='B1F3DF2C7803B02C326BAB74BB140979';



select distinct query\_sql from gv$sql\_audit where sql\_id='F594DCAC2224F0B0976DDD6A8525E6CB'\G;



##### 5.3 分析执行计划

obclient -h 172.16.100.21 -utpcc@test\_tenant -P2883 -p'password' -c -A tpcc



explain SELECT c\_data    FROM bmsql\_customer    WHERE c\_w\_id = 1 AND c\_d\_id = 9 AND c\_id = 511;\G



explain UPDATE bmsql\_district    SET d\_ytd = d\_ytd + 479.86    WHERE d\_w\_id = 1 AND d\_id = 10\G;



explain SELECT d\_name, d\_street\_1, d\_street\_2, d\_city,       d\_state, d\_zip    FROM bmsql\_district    WHERE d\_w\_id = 1 AND d\_id = 10\G;



explain UPDATE bmsql\_warehouse    SET w\_ytd = w\_ytd + 479.86    WHERE w\_id = 1\G;



explain SELECT w\_name, w\_street\_1, w\_street\_2, w\_city,       w\_state, w\_zip    FROM bmsql\_warehouse    WHERE w\_id = 1 \G;



explain SELECT c\_id    FROM bmsql\_customer    WHERE c\_w\_id = 1 AND c\_d\_id = 10 AND c\_last = 'ANTIPRESABLE'    ORDER BY c\_first\G;



explain SELECT c\_first, c\_middle, c\_last, c\_street\_1, c\_street\_2,       c\_city, c\_state, c\_zip, c\_phone, c\_since, c\_credit,       c\_credit\_lim, c\_discount, c\_balance    FROM bmsql\_customer    WHERE c\_w\_id = 1 AND c\_d\_id = 10 AND c\_id = 643    FOR UPDATE\G;



explain UPDATE bmsql\_customer    SET c\_balance = c\_balance - 479.86,        c\_ytd\_payment = c\_ytd\_payment + 479.86,        c\_payment\_cnt = c\_payment\_cnt + 1    WHERE c\_w\_id = 1 AND c\_d\_id = 10 AND c\_id = 643\G;



explain INSERT INTO bmsql\_history (   h\_c\_id, h\_c\_d\_id, h\_c\_w\_id, h\_d\_id, h\_w\_id,    h\_date, h\_amount, h\_data) VALUES (643, 10, 1, 10, 1, '2022-02-07 09:09:09.087', 479.86, 'seWDa94k   x2iIRI')\G;

