# 1 BenchmarkSQL运行

## 1.1配置BenchmarkSQL

下载BenchmarkSQL，并解压

wget <https://jaist.dl.sourceforge.net/project/benchmarksql/benchmarksql-5.0.zip>

[root@oceanbase ~]# unzip benchmarksql-5.0.zip

下载JDBC驱动

[OceanBase 公有云版下载 - 云数据库工具下载 -OceanBase 数据库下载中心](https://www.oceanbase.com/softwarecenter-cloud)

[root@oceanbase ~]# mv oceanbase-client-2.4.1.jar /root/benchmarksql-5.0/lib/oracle

[root@oceanbase ~]# cd /root/benchmarksql-5.0/lib/oracle

[root@oceanbase oracle]# chmod 777 oceanbase-client-2.4.1.jar

安装ant，并执行

[root@oceanbase oracle]# yum install -y ant



编辑配置文件

[root@oceanbase run]# vi props.ob

db=oracle

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

conn=jdbc:oceanbase://127.0.0.1:2883/tpcc?useUnicode=true&characterEncoding=utf-8

user=root

password=rootroot

db=oracle

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

conn=jdbc:oceanbase://127.0.0.1:2883/tpcc?useUnicode=true&characterEncoding=utf-8

user=root

password=Zy1441128

warehouses=2

loadWorkers=1

terminals=5

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

runTxnsPerTerminal=0

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

runMins=5

//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

~

创建对应数据库

obclient -h 192.168.6.31 -P 2883 -uroot -pZy1441128 -c -A

create database tpcc;



## 1.2 运行TPCC

运行脚本

[root@oceanbase run]# ./runDatabaseBuild.sh props.ob



由于是oracle语法，oracle实现自增列用的是序列+触发器。oceanbase语法不支持。我们直接手动创建自增列即可

ALTER TABLE bmsql\_history MODIFY hist\_id INTEGER AUTO\_INCREMENT PRIMARY KEY;

如果不创建在跑runBenchmark.sh时会报如下错误

Caused by: java.sql.SQLException: Field 'hist\_id' doesn't have a default value



开始TPCC测试

./runBenchmark.sh props.ob



# 2 查看执行计划

查看 processlist 确认有会话连接到集群

show full process list;



查看 TOP SQL,使用OceanBase数据库

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

FROM gv$ob\_sql\_audit s

WHERE 1=1

and db\_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 3;



根据 sql\_id 定位具体的 sql 语句 ，并查询 SVR\_IP,SVR\_PORT,TENANT\_ID,PLAN\_ID 用于获取真实实际的执行计划

select SVR\_IP,SVR\_PORT,TENANT\_ID,PLAN\_ID,QUERY\_SQL from gv$ob\_sql\_audit where sql\_id='F59A700FA168324279B0DBC25E19760F' limit 1;



查看执行计划

obclient [oceanbase]> use tpcc

obclient [tpcc]> explain SELECT count(\*) AS low\_stock FROM ( SELECT s\_w\_id, s\_i\_id, s\_quantity FROM bmsql\_stock WHERE s\_w\_id = 2 AND s\_quantity < 16 AND s\_i\_id IN ( SELECT ol\_i\_id FROM bmsql\_district JOIN bmsql\_order\_line ON ol\_w\_id = d\_w\_id AND ol\_d\_id = d\_id AND ol\_o\_id >= d\_next\_o\_id - 20 AND ol\_o\_id < d\_next\_o\_id WHERE d\_w\_id = 2 AND d\_id = 8 ) );

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

| Query Plan |

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

| ==================================================================== |

| |ID|OPERATOR |NAME |EST.ROWS|EST.TIME(us)| |

| -------------------------------------------------------------------- |

| |0 |SCALAR GROUP BY | |1 |44 | |

| |1 |└─HASH RIGHT SEMI JOIN | |4 |44 | |

| |2 | ├─SUBPLAN SCAN |VIEW1 |4 |16 | |

| |3 | │ └─NESTED-LOOP JOIN | |4 |16 | |

| |4 | │ ├─TABLE RANGE SCAN|bmsql\_order\_line|73 |12 | |

| |5 | │ └─MATERIAL | |1 |3 | |

| |6 | │ └─TABLE GET |bmsql\_district |1 |3 | |

| |7 | └─TABLE RANGE SCAN |bmsql\_stock |98 |19 | |

| ==================================================================== |

| Outputs & filters: |

| ------------------------------------- |

| 0 - output([T\_FUN\_COUNT(\*)]), filter(nil), rowset=256 |

| group(nil), agg\_func([T\_FUN\_COUNT(\*)]) |

| 1 - output(nil), filter(nil), rowset=256 |

| equal\_conds([bmsql\_stock.s\_i\_id = VIEW1.ol\_i\_id]), other\_conds(nil) |

| 2 - output([VIEW1.ol\_i\_id]), filter(nil), rowset=256 |

| access([VIEW1.ol\_i\_id]) |

| 3 - output([bmsql\_order\_line.ol\_i\_id]), filter(nil), rowset=256 |

| conds([bmsql\_order\_line.ol\_o\_id < bmsql\_district.d\_next\_o\_id], [bmsql\_order\_line.ol\_o\_id >= bmsql\_district.d\_next\_o\_id - 20]), nl\_params\_(nil), use\_batch=false |

| 4 - output([bmsql\_order\_line.ol\_o\_id], [bmsql\_order\_line.ol\_i\_id]), filter(nil), rowset=256 |

| access([bmsql\_order\_line.ol\_o\_id], [bmsql\_order\_line.ol\_i\_id]), partitions(p0) |

| is\_index\_back=false, is\_global\_index=false, |

| range\_key([bmsql\_order\_line.ol\_w\_id], [bmsql\_order\_line.ol\_d\_id], [bmsql\_order\_line.ol\_o\_id], [bmsql\_order\_line.ol\_number]), range(2,8,MIN,MIN ; 2, |

| 8,MAX,MAX), |

| range\_cond([bmsql\_order\_line.ol\_w\_id = 2], [bmsql\_order\_line.ol\_d\_id = 8]) |

| 5 - output([bmsql\_district.d\_next\_o\_id], [bmsql\_district.d\_next\_o\_id - 20]), filter(nil), rowset=256 |

| 6 - output([bmsql\_district.d\_next\_o\_id], [bmsql\_district.d\_next\_o\_id - 20]), filter([bmsql\_district.d\_next\_o\_id > bmsql\_district.d\_next\_o\_id - 20]), rowset=256 |

| access([bmsql\_district.d\_next\_o\_id]), partitions(p0) |

| is\_index\_back=false, is\_global\_index=false, filter\_before\_indexback[false], |

| range\_key([bmsql\_district.d\_w\_id], [bmsql\_district.d\_id]), range[2,8 ; 2,8], |

| range\_cond([bmsql\_district.d\_w\_id = 2], [bmsql\_district.d\_id = 8]) |

| 7 - output([bmsql\_stock.s\_i\_id]), filter([bmsql\_stock.s\_quantity < 16]), rowset=256 |

| access([bmsql\_stock.s\_i\_id], [bmsql\_stock.s\_quantity]), partitions(p0) |

| is\_index\_back=false, is\_global\_index=false, filter\_before\_indexback[false], |

| range\_key([bmsql\_stock.s\_w\_id], [bmsql\_stock.s\_i\_id]), range(2,MIN ; 2,MAX), |

| range\_cond([bmsql\_stock.s\_w\_id = 2]) |

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

39 rows in set (0.066 sec)

# 查看真实的执行计划

根据上面查到的 SVR\_IP,SVR\_PORT,TENANT\_ID,PLAN\_ID 查询 GV$OB\_PLAN\_CACHE\_PLAN\_EXPLAIN以获取真实实际的执行计划

obclient [oceanbase]> select \* from GV$OB\_PLAN\_CACHE\_PLAN\_EXPLAIN where SVR\_IP = '192.168.6.31' and SVR\_PORT=2882 and TENANT\_ID=1 and PLAN\_ID=5911;

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

| TENANT\_ID | SVR\_IP | SVR\_PORT | PLAN\_ID | PLAN\_DEPTH | PLAN\_LINE\_ID | OPERATOR | NAME | ROWS | COST | PROPERTY |

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

| 1 | 192.168.6.31 | 2882 | 5911 | 0 | 0 | PHY\_SCALAR\_AGGREGATE | NULL | 1 | 39 | NULL |

| 1 | 192.168.6.31 | 2882 | 5911 | 1 | 1 | PHY\_NESTED\_LOOP\_JOIN | NULL | 2 | 39 | NULL |

| 1 | 192.168.6.31 | 2882 | 5911 | 2 | 2 | PHY\_SUBPLAN\_SCAN | NULL | 2 | 10 | NULL |

| 1 | 192.168.6.31 | 2882 | 5911 | 3 | 3 | PHY\_HASH\_DISTINCT | NULL | 2 | 10 | NULL |

| 1 | 192.168.6.31 | 2882 | 5911 | 4 | 4 | PHY\_NESTED\_LOOP\_JOIN | NULL | 2 | 9 | NULL |

| 1 | 192.168.6.31 | 2882 | 5911 | 5 | 5 | PHY\_TABLE\_SCAN | bmsql\_order\_line | 28 | 6 | table\_rows:379334, physical\_range\_rows:28, logical\_range\_rows:28, index\_back\_rows:0, output\_rows:28, avaiable\_index\_name[bmsql\_order\_line] |

| 1 | 192.168.6.31 | 2882 | 5911 | 5 | 6 | PHY\_MATERIAL | NULL | 1 | 2 | NULL |

| 1 | 192.168.6.31 | 2882 | 5911 | 6 | 7 | PHY\_TABLE\_SCAN | bmsql\_district | 1 | 2 | table\_rows:20, physical\_range\_rows:1, logical\_range\_rows:1, index\_back\_rows:0, output\_rows:1, avaiable\_index\_name[bmsql\_district] |

| 1 | 192.168.6.31 | 2882 | 5911 | 2 | 8 | PHY\_TABLE\_SCAN | bmsql\_stock | 1 | 20 | table\_rows:100049, physical\_range\_rows:1, logical\_range\_rows:1, index\_back\_rows:0, output\_rows:0, avaiable\_index\_name[bmsql\_stock] |

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

9 rows in set (0.001 sec)