实验三报告

## 关卡一：openGauss数据库的编译和安装

### 关卡验证

步骤1 首先需要对数据库状态进行验证。

[omm@opengauss01 openGauss-server]$ **gs\_ctl status**

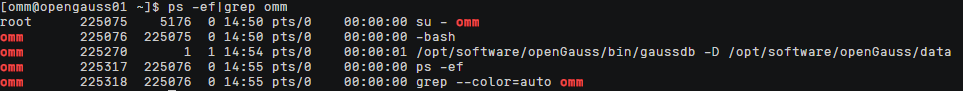
（截图语句和执行结果）



步骤2 对数据库进程进行截图验证，需包含数据库服务器的主机名。

[omm@opengauss01 openGauss-server]$ **ps -ef|grep omm**

（截图语句和执行结果）



## 关卡二：openGauss数据导入及基本操作

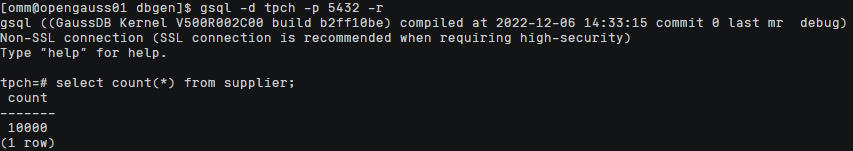
### 关卡验证

步骤12 登录数据库验证

[omm@opengauss01 dbgen]$ **gsql -d tpch -p 5432 -r**

tpch=# **select count(\*) from supplier;**

（截图语句和执行结果）

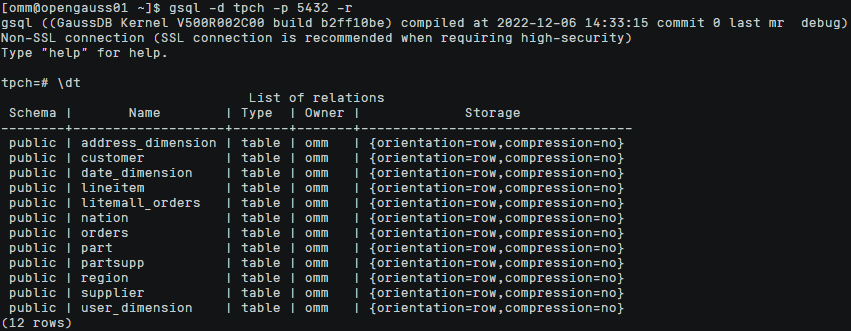


步骤21 登录数据库进行验证

[omm@opengauss01 ~]$ **gsql -d tpch -p 5432 -r**

tpch=# **\dt**

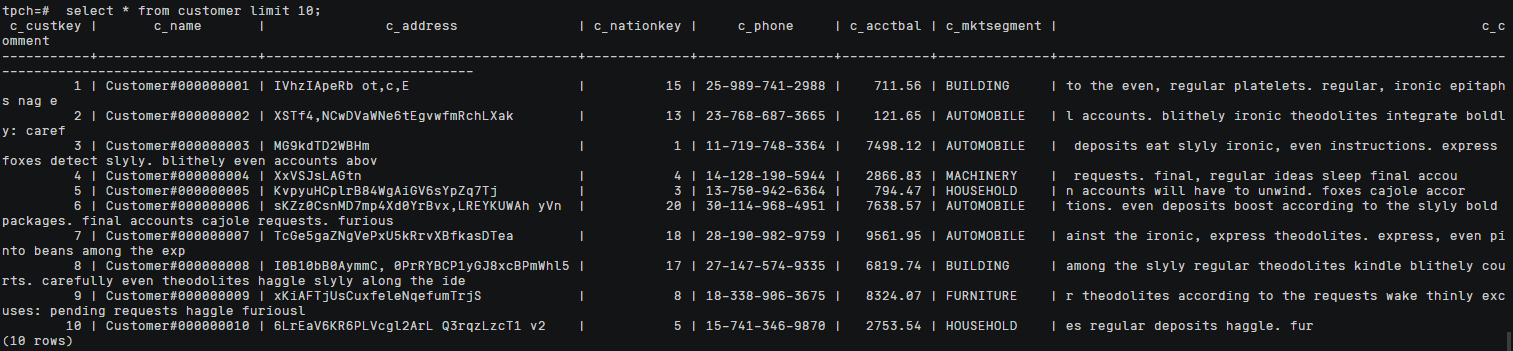
（截图语句和执行结果）



步骤22 查询customer表的数据

tpch=# **select \* from customer limit 10;**

（截图语句和执行结果）



### 思考题

数据初始化中出现了TPC-H，这是什么？

TPC-H（商业智能计算测试） 是美国交易处理效能委员会（TPC,Transaction Processing Performance Council) 组织制定的用来模拟决策支持类应用的一个测试集。目前，在学术界和工业界普遍用来评价决策支持技术方面应用的性能。

## 关卡三：openGauss的AI4DB特性应用

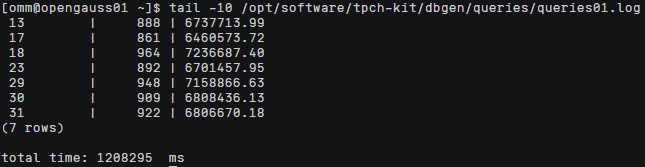
### 关卡验证

（1）使用X-Tuner进行参数优化

步骤2 在原来CloudShell连接窗口中查看queries01.log。

[omm@opengauss01 ~]$ **tail -10 /opt/software/tpch-kit/dbgen/queries/queries01.log**

（截图执行语句和结果）

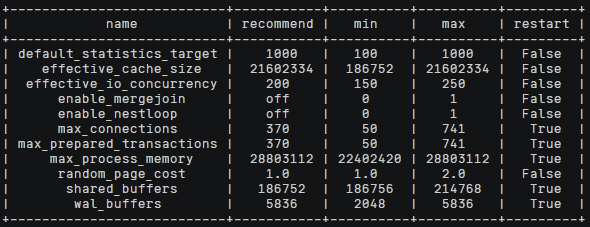


步骤3 切换至root用户，执行X-Tuner进行参数建议优化

[omm@opengauss01 ~]$ **exit**

[root@opengauss01 xtuner]# **gs\_xtuner recommend --db-name tpch --db-user omm --port 5432 --host 127.0.0.1 --host-user omm**

（截图执行语句和结果）

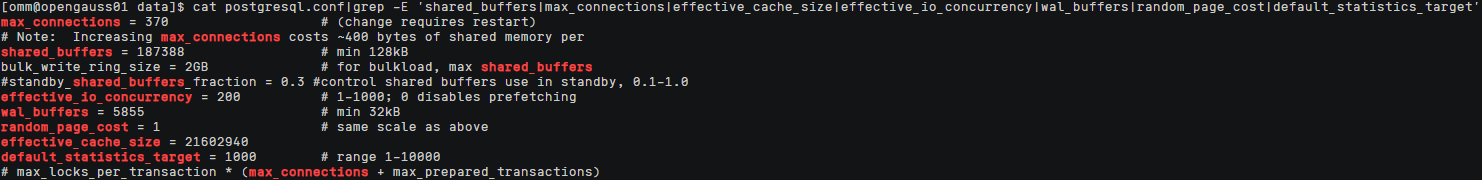


步骤6 获取参数值

[omm@opengauss01 ~]$ **cd /opt/software/openGauss/data**

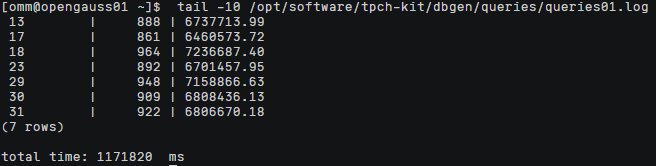
[omm@opengauss01 data]$ **cat postgresql.conf|grep -E 'shared\_buffers|max\_connections|effective\_cache\_size|effective\_io\_concurrency|wal\_buffers|random\_page\_cost|default\_statistics\_target'**

（截图执行语句和结果）



步骤7 再次执行步骤2，对比优化前的执行时间。

（截图执行语句和结果）



步骤8 【附加题】有兴趣的同学可以尝试并截图记录于此。

（截图执行语句和结果）

（2）Index-advisor：索引推荐

步骤4 使用explain，对该SQL加以分析

tpch=# **EXPLAIN**

**SELECT ad.province AS province, SUM(o.actual\_price) AS GMV**

**FROM litemall\_orders o,**

**address\_dimension ad,**

**date\_dimension dd**

**WHERE o.address\_key = ad.address\_key**

**AND o.add\_date = dd.date\_key**

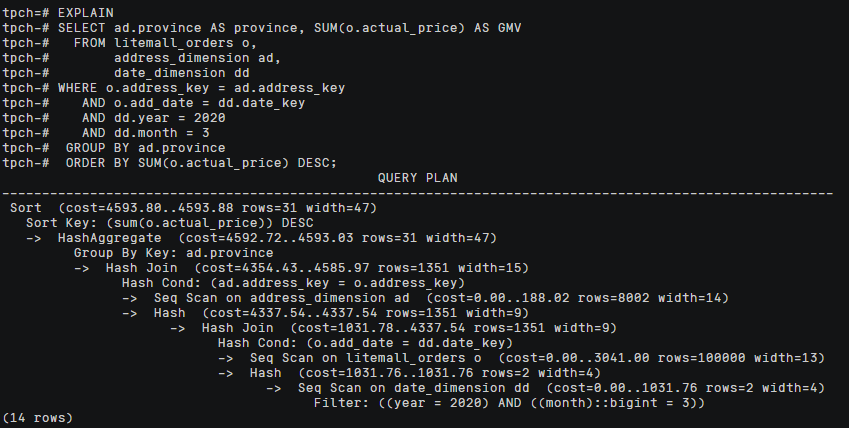
**AND dd.year = 2020**

**AND dd.month = 3**

**GROUP BY ad.province**

**ORDER BY SUM(o.actual\_price) DESC;**

（截图执行语句和结果）



步骤10 使用explain，对该SQL加以分析

tpch=# **EXPLAIN**

**SELECT ad.province AS province, SUM(o.actual\_price) AS GMV**

**FROM litemall\_orders o,**

**address\_dimension ad,**

**date\_dimension dd**

**WHERE o.address\_key = ad.address\_key**

**AND o.add\_date = dd.date\_key**

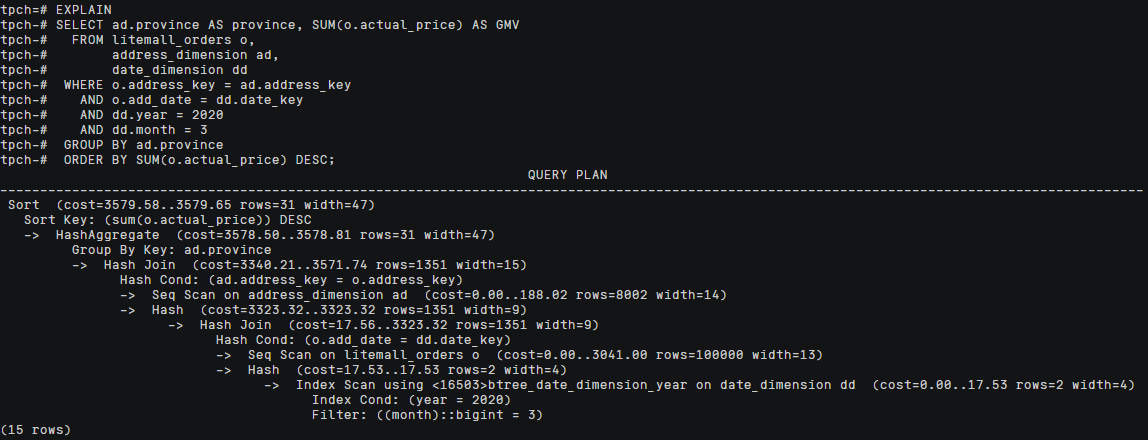
**AND dd.year = 2020**

**AND dd.month = 3**

**GROUP BY ad.province**

**ORDER BY SUM(o.actual\_price) DESC;**

（截图执行语句和结果）



步骤11 【附加题】有兴趣的同学可以尝试并截图记录于此。仅需要从queries.sql文件里选择一条或多条进行索引优化即可。

（截图执行语句和结果）

## 关卡四【附加题】：openGauss的DB4AI特性应用

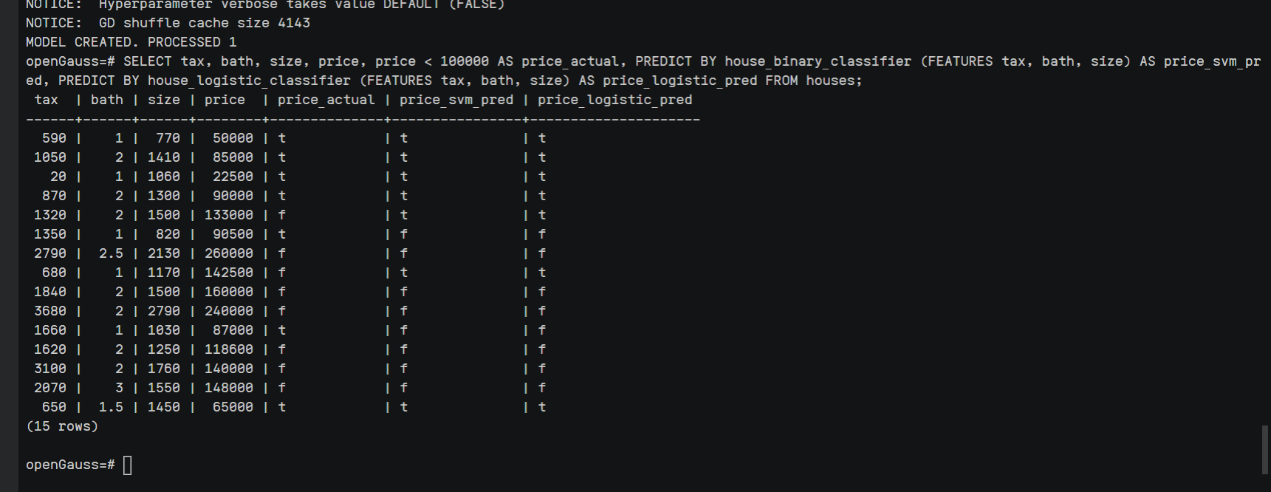
\*本关卡为附加题，有兴趣的同学可以尝试实验并记录于此。

### 关卡验证

步骤10 利用训练好的逻辑回归模型预测数据，并与SVM算法进行比较，将执行结果截图。

openGauss=# **SELECT tax, bath, size, price, price < 100000 AS price\_actual, PREDICT BY house\_binary\_classifier (FEATURES tax, bath, size) AS price\_svm\_pred, PREDICT BY house\_logistic\_classifier (FEATURES tax, bath, size) AS price\_logistic\_pred FROM houses;**

（截图执行语句和结果）



## 清理工作：资源释放

### 关卡验证

步骤3 查看到列表中已没有资源时，表示弹性云服务器已删除。

（截图执行语句和结果）

