生成 TPCH 数据并导入到 Hive

TPC-H是事务处理性能委员会（ Transaction ProcessingPerformance Council ）制定的基准程序之一，TPC- H 主要目的是评价特定查询的决策支持能力，该基准模拟了决策支持系统中的数据库操作，测试数据库系统复杂查询的响应时间，以每小时执行的查询数(TPC-H QphH@Siz)作为度量指标。

我们在很多大数据系统上线或者产品上线的时候一般都会测试一下 TPCH 的性能。测试 TPCH 的时候一般都需要有数据，本文将给大家介绍一下如何生成 TPCH 数据，并导入到 Hive。

下载 TPCH 数据生成代码

TPC
官方网站其实就给我们准备了生成数据的
代码，可以到这里寻找最新的数据，或者你直接点击下面表格中对应的 TPC 测试。

Benchmark/Document	Current Version	Specification	Source Code
TPC-C	5.11.0	pdf	n/a
TPC-DI	1.1.0	pdf	Download TPC-DI_Tools_v1.1.0.zip
TPC-DS	3.2.0	pdf	Download TPC-DS_Tools_v3.2.0.zip
TPC-E	1.14.0	pdf	Download TPC-E_Tools_v1.14.0.zip
TPC-H	3.0.0	pdf	Download TPC-H_Tools_v3.0.0.zip
TPCX-AI	1.0.0	pdf	Download TPCX-AI_Tools_v1.0.0.zip
TPCX-BB	1.5.1	pdf	Download TPCX-BB_Tools_v1.5.1.zip
TPCX-HCI	1.1.8	pdf	Download TPCx-HCI_Benchmarking_Kit_v1.1.8.zip
TPCX-HS	2.0.3	pdf	Download TPCX-HS_Tools_v2.0.3.zip

TPCX-IOT	2.0.0	pdf	Download TPCx-iot_Tools_v2.0.0.zip
TPCX-V	2.1.8	pdf	Download TPCx-V_Benchmarking_Kit_v2.1.8.zip

修改数据生成参数并编译代码

上面步骤下载完 TPC-H_Tools 之后，我们解压可以得到一个名为 TPC-H_Tools_v3.0.0 的目录，我们进入这个目录，并进入到 dbgen 目录里面，找到 makefile.suite 文件，并按照下面修改相关配置：

```
#################
## CHANGE NAME OF ANSI COMPILER HERE
#################
CC      = gcc
# Current values for DATABASE are: INFORMIX, DB2, TDAT (Teradata)
#                                  SQLSERVER, SYBASE, ORACLE, VECTORWISE
# Current values for MACHINE are:  ATT, DOS, HP, IBM, ICL, MVS,
#                                  SGI, SUN, U2200, VMS, LINUX, WIN32
# Current values for WORKLOAD are:  TPCH
DATABASE= HIVE
MACHINE = LINUX
WORKLOAD = TPCH
```

注意，TPC-H_Tools 的 DATABASE 并没有给我们提供 HIVE 这种数据库，所以我们按照上面修改完 makefile.suite 文件之后，我们还需要修改 tpcd.h 文件，并在 #ifdef ORACLE 前面加上下面配置

```
#ifdef  HIVE
#define GEN_QUERY_PLAN  "explain;"
#define START_TRAN      "start transaction;₩n"
#define END_TRAN        "commit;₩n"
#define SET_OUTPUT      ""
#define SET_ROWCOUNT    "limit %d;₩m"
#define SET_DBASE       "use %s;₩n"
#endif
```

修改完之后，我们就可以保存，然后使用下面命令编译代码：

make -f makefile.suite

生成 TPCH 数据

编译完代码之后，会在当前目录下生成 dbgen 和 qgen 两份文件。我们可以使用 dbgen 生成 TPCH 测试数据集：

./dbgen -s 1 -f

其中 -s 代表 Scale Factor（简称 SF），也就是测试数据集的规模大小，默认为1，1 个 SF 约生成 GB 的数据。-f 代表强制覆盖之前生成的文件。更多的使用可以通过 ./dbgen --help 了解。

运行完上面命令之后，就可以在本地生成如下文件：

```
 dbgen ls -lh *.tbl
-rw-r--r--  1 iteblog  iteblog    23M 11 29 22:09 customer.tbl
-rw-r--r--  1 iteblog  iteblog   725M 11 29 22:09 lineitem.tbl
-rw-r--r--  1 iteblog  iteblog   2.2K 11 29 22:09 nation.tbl
-rw-r--r--  1 iteblog  iteblog   164M 11 29 22:09 orders.tbl
-rw-r--r--  1 iteblog  iteblog    23M 11 29 22:09 part.tbl
-rw-r--r--  1 iteblog  iteblog   113M 11 29 22:09 partsupp.tbl
-rw-r--r--  1 iteblog  iteblog   389B 11 29 22:09 region.tbl
-rw-r--r--  1 iteblog  iteblog   1.3M 11 29 22:09 supplier.tbl
```

注意，生成的数据默认分隔符是 |，大家可以修改 dss.h 文件里面的 #define SEPARATOR '|' 来指定其他分隔符。

将 TPCH 数据集导入到 HIVE

测试数据生成之后，我们可以创建相关的表：

```
create database tpch;
use tpch;

create external table lineitem (
  l_orderkey int,
  l_partkey int,
```

```
  l_suppkey int,
  l_linenumber int,
  l_quantity double,
  l_extendedprice double,
  l_discount double,
  l_tax double,
  l_returnflag string,
  l_linestatus string,
  l_shipdate string,
  l_commitdate string,
  l_receiptdate string,
  l_shipinstruct string,
  l_shipmode string,
  l_comment string)
row format delimited
fields terminated by '|'
stored as textfile;

create external table nation (
  n_nationkey int,
  n_name string,
  n_regionkey int,
  n_comment string)
row format delimited
fields terminated by '|'
stored as textfile;

create external table region (
  r_regionkey int,
  r_name string,
  r_comment string)
row format delimited
fields terminated by '|'
stored as textfile;

create external table part (
  p_partkey int,
  p_name string,
  p_mfgr string,
  p_brand string,
  p_type string,
  p_size int,
  p_container string,
  p_retailprice double,
  p_comment string)
row format delimited
```

```
fields terminated by '|'
stored as textfile;

create external table supplier (
  s_suppkey int,
  s_name string,
  s_address string,
  s_nationkey int,
  s_phone string,
  s_acctbal double,
  s_comment string)
row format delimited
fields terminated by '|'
stored as textfile;

create external table partsupp (
  ps_partkey int,
  ps_suppkey int,
  ps_availqty int,
  ps_supplycost double,
  ps_comment string)
row format delimited
fields terminated by '|'
stored as textfile;

create external table customer (
  c_custkey int,
  c_name string,
  c_address string,
  c_nationkey int,
  c_phone string,
  c_acctbal double,
  c_mktsegment string,
  c_comment string)
row format delimited
fields terminated by '|'
stored as textfile;

create external table orders (
  o_orderkey int,
  o_custkey int,
  o_orderstatus string,
  o_totalprice double,
  o_orderdate date,
  o_orderpriority string,
  o_clerk string,
```

```
  o_shippriority int,
  o_comment string)
row format delimited
fields terminated by '|'
stored as textfile;
```

然后通过下面命令把数据导入到对应的 Hive 表：

```
LOAD DATA LOCAL INPATH '/home/iteblog/tpch/tpc_h_tool/dbgen/region.tbl' INTO TABLE regi
on;
LOAD DATA LOCAL INPATH '/home/iteblog/tpch/tpc_h_tool/tpc_h_tool/dbgen/nation.tbl' INTO
TABLE nation;
LOAD DATA LOCAL INPATH '/home/iteblog/tpch/tpc_h_tool/tpc_h_tool/dbgen/part.tbl'   INTO T
ABLE part;
LOAD DATA LOCAL INPATH '/home/iteblog/tpch/tpc_h_tool/tpc_h_tool/dbgen/supplier.tbl' INT
O TABLE supplier;
LOAD DATA LOCAL INPATH '/home/iteblog/tpch/tpc_h_tool/tpc_h_tool/dbgen/partsupp.tbl' INT
O TABLE partsupp;
LOAD DATA LOCAL INPATH '/home/iteblog/tpch/tpc_h_tool/tpc_h_tool/dbgen/customer.tbl' IN
TO TABLE customer;
LOAD DATA LOCAL INPATH '/home/iteblog/tpch/tpc_h_tool/tpc_h_tool/dbgen/orders.tbl' INTO
TABLE orders;
LOAD DATA LOCAL INPATH '/home/iteblog/tpch/tpc_h_tool/tpc_h_tool/dbgen/lineitem.tbl' INT
O TABLE lineitem;
```

之后，我们就可以测试 22 条 TPCH SQL 了。关于 TPCH SQL 可以参见这里：《TPCH SQL 含义解析》