Flink SQL Benchmark设计

## TPC-H简介

TPC-H 基准测试是由 TPC-D(由 TPC 组织于 1994 年指定的标准,用于决策支持系统方面的测试基准)发展而来的.TPC-H 用 3NF 实现了一个数据仓库,共包含 8 个基本关系,其数据量可以设定从 1G~3T 不等。TPC-H 基准测试包括 22 个查询(Q1~Q22),其主要评价指标是各个查询的响应时间,即从提交查询到结果返回所需时间.TPC-H 基准测试的度量单位是每小时执行的查询数( QphH@size)，其中 H 表示每小时系统执行复杂查询的平均次数，size 表示数据库规模的大小,它能够反映出系统在处理查询时的能力。

**表结构**

TPC-H包括8张表（表上有些约束等需要满足）

PART：表示零件的信息

SUPPLIER：表示供货商的信息

PARTSUPP：表示供货商的零件的信息。

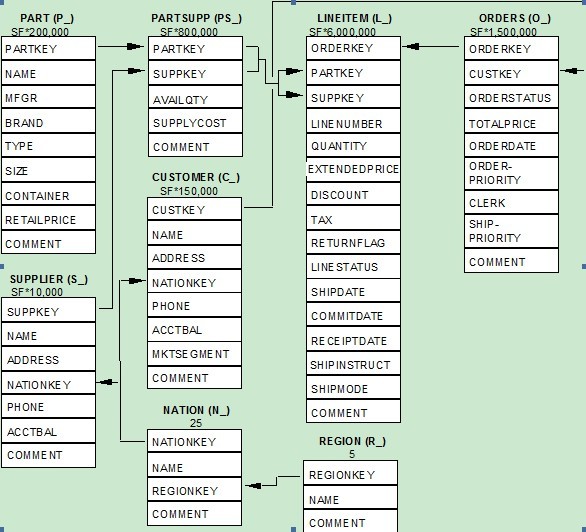
CUSTOMER：表示消费者的信息。

ORDERS：表示订单的信息。

LINEITEM：表示在线商品的信息。

NATION：表示国家的信息。

REGION：表示地区的信息。



SF指scale factor，即数据规模，可选1、10、30、100、300、1 000、3 000、10 000

**参数特征**

|  |  |  |
| --- | --- | --- |
| 选项 | 参数 | 作用 |
| -f | None | 覆盖已存在的文件 |
| -s | <scale> | 数据规模  （1,10,30,100,1000,3000,10000） |
| -T | <table> | 单独生成某个table |
| -C | <children> | 数据分成children份生成 |
| -S | <n> | 生成第n份数据 |

由于 -C 和 –S选项的支持，我们可以分布式的生成数据，比如将table分成num\_file份，放到不同的slave节点，分别生成，然后上传到HDFS中。

./dbgen –f –s <scale\_factor> -C <num\_files> -S<nth\_part\_file>

**查询语句与特征**

TPC-H共设计了22个select语句，遵守SQL-92语法

|  |  |  |
| --- | --- | --- |
| 语句 | 特征 | 相关表数目 |
| Q01 | 统计查询 | 1 |
| Q02 | WHERE条件中，使用子查询(=) | 5 |
| Q03 | 多表关联统计查询，并统计(SUM) | 3 |
| Q04 | WHERE条件中，使用子查询(EXISTS)，并统计(COUNT) | 2 |
| Q05 | 多表关联查询(=)，并统计(SUM) | 6 |
| Q06 | 条件(BETWEEN AND)查询，并统计(SUM) | 1 |
| Q07 | 带有FROM子查询，从结果集中统计(SUM) | 5 |
| Q08 | 带有FROM多表子查询，从结果集中的查询列上带有逻辑判断(WHEN THEN ELSE)的统计(SUM) | 7 |
| Q09 | 带有FROM多表子查询，查询表中使用函数(EXTRACT)，从结果集中统计(SUM) | 6 |
| Q10 | 多表条件查询(>=, <)，并统计(SUM) | 4 |
| Q11 | 在GROUP BY中使用比较条件(HAVING >)，比较值从子查询中查出 | 4 |
| Q12 | 带有逻辑判断(WHEN AND/ WHEN OR)的查询，并统计(SUM) | 2 |
| Q13 | 带有FROM子查询，子查询中使用外联结 | 2 |
| Q14 | 使用逻辑判断(WHEN ELSE)的查询 | 2 |
| Q15 | 使用视图和表关联查询 | 2 |
| Q16 | 在WHERE子句中使用子查询，使用IN/ NOT IN判断条件，并统计(COUNT) | 4 |
| Q17 | 在WHERE子句中使用子查询，使用<比较，使用了AVG函数 | 2 |
| Q18 | 在WHERE子句中使用IN条件从子查询结果中比较 | 3 |
| Q19 | 多条件比较查询 | 2 |
| Q20 | WHERE条件子查询(三层) | 5 |
| Q21 | 在WHERE条件中使用子查询，使用EXISTS和NOT EXISTS判断 | 4 |
| Q22 | 在WHERE条件中使用判断子查询、IN、NOT EXISTS，并统计(SUM、COUNT)查询结果 | 2 |

## 数据生成模块

数据生成的流程



Master上脚本gendata.py

参数：

|  |  |
| --- | --- |
| scale\_factor | 要生成数据规模 |
| num\_files | 生成表的份数 |
| host\_list | 参与数据生成的主机 |
| local\_dir | 数据集生成工具位置 |
| Hdfs\_path | Hdfs位置 |

1.参数验证

2.在HDFS\_PATH上建目录

3.删除原始的数据集

4.计算每个node上需要生成的数据集份数

5.建立多个线程，在每个node上运行gen\_per\_node.py脚本

6.等待全部线程完成后结束

Slave node 上脚本 gen\_per\_node.py

参数：

|  |  |
| --- | --- |
| scale\_factor | 要生成数据规模 |
| num\_files | 生成表的份数 |
| first\_index | 第First\_index份 |
| last\_index | 第last\_index份 |
| Hdfs\_path | Hdfs位置 |

1.make编译tpch数据生成脚本

2.运行生成数据集在本地

3.上传数据集到hdfs上

4.删除本地的数据集

5.结束线程

## 执行语句模块

执行流程如下：

1. 获取执行的query序号
2. 获取当前query相关的表
3. 依次加载相关表，注册到环境中
4. 执行查询语句
5. 打印结果

## 初步测试结果

1G数据集，Flink1.2

|  |  |
| --- | --- |
| Q1 | \ |
| Q2 | 多表连接问题 |
| Q3 | \ |
| Q4 | \ |
| Q5 | \ |
| Q6 | \ |
| Q7 | \ |
| Q8 | 多表连接问题 |
| Q9 | 多表连接问题 |
| Q10 | \ |
| Q11 | \ |
| Q12 | \ |
| Q13 | \ |
| Q14 | \ |
| Q15 | drop,view不支持 |
| Q16 | distinct不支持 |
| Q17 | flink节点转化问题,不等连接不支持 |
| Q18 | \ |
| Q19 | 多表连接问题 |
| Q20 | \ |
| Q21 | flink节点转化问题, not exist不支持 |
| Q22 | flink节点转化问题, not exist不支持 |

多表连接问题：

flink目前不支持笛卡尔连接/无条件join。

当flink进行多表连接时，虽然可以通过调整join顺序和过滤操作配对来使得不进行笛卡尔连接，但是calcite解析第一步时会按照给定的join顺序生成未优化的逻辑计划，这时就有可能是笛卡尔连接操作，之后flink在转换逻辑计划时就会报不支持的异常。

Flink节点转化问题：

calcite逻辑计划在优化和转换成Flink节点时,某些规则（DataSetJoin）可能会报错，无法转换