实验三 基于 flinksql 的流式数据查询

22121630 汪江豪

实验步骤及结果：

1. 创建基于实验二制作的镜像，先下载 java 运行环境和 java 开发工具包
   sudo apt-get install openjdk-8-jre openjdk-8-jdk
2. 将 java 添加到环境变量
   vim ~/bashrc
   第一行添加
   export JAVA_HOME=/usr/local/lib/jvm/java-8-openjdk-amd64
   保存退出后，应用修改
   source ~/.bashrc

3.输出 java 版号以验证 java 成功安装

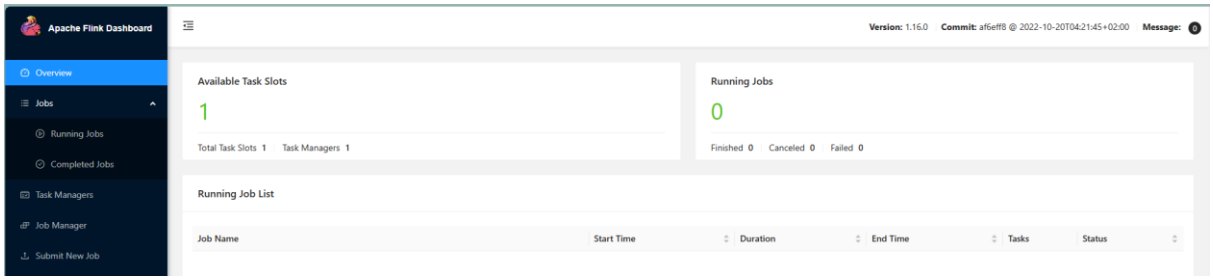
```
ubuntu@VM-0-13-ubuntu:~$ java -version
openjdk version "1.8.0_442"
OpenJDK Runtime Environment (build 1.8.0_442-8u442-b06~us1-0ubuntu1~20.04-b06)
OpenJDK 64-Bit Server VM (build 25.442-b06, mixed mode)
```

4.安装 flink
   wget https://archive.apache.org/dist/flink/flink-1.16.0/flink-1.16.0-bin-scala_2.12.tgz
   解压并改名
   sudo tar -zxvf flink-1.16.0-bin-scala_2.12.tgz -C /usr/local
   cd /usr/local
   sudo mv flink-1.16.0 flink
   添加 flink 主路径到环境变量并应用
   vim ~/.bashrc
   export FLINK_HOME=/usr/local/flink
   export PATH=$FLINK_HOME/bin:$PATH
   source ~/.bashrc

5. 到 /usr/local/flink/conf 目录下修改 flink-conf.yaml 文件，将 rest.address 和 rest.bind-address 改为 0.0.0.0

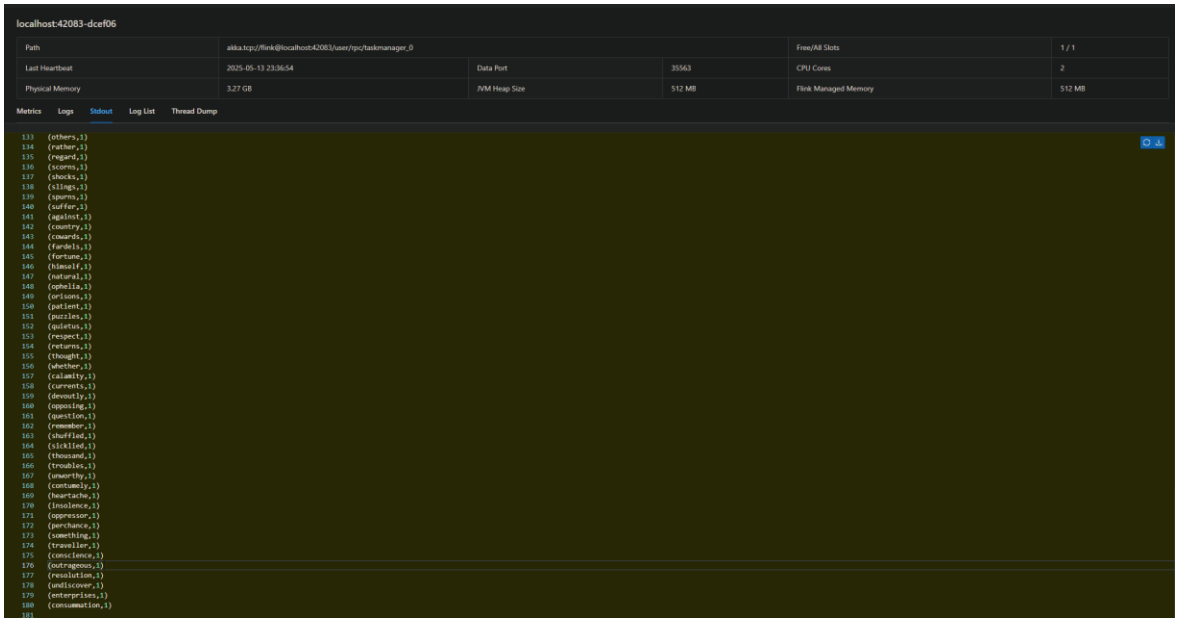
6.到腾讯云安全组界面放开 8081 端口，启动 flink 集群
   start-cluster.sh(路径已添加到环境变量，所以可以这么写)

7.在任何一个浏览器中输入[服务器的公网 ip]:8081，可进入到 flinksql 的 webui 界面，如下图：



8.尝试提交一个示例文件，在 webui 的 taskmanager 下查看结果如下

flink run /usr/local/flink/examples/batch/WordCount.jar



9. 安装 maven 打包编译工具

wget https://dlcdn.apache.org/maven/maven-3/3.9.4/binaries/apache-maven-3.9.4-bin.zip

也可以将 maven 主目录同样添加到环境变量，便于打包编译 java 项目。

10. 新建 tableapp 文件夹和 tableapp/src/main/java 目录，用于存储项目的作业代码，并写好 pom.xml 配置文件。

11. 导入 medium_relation 数据集后，通过写 java 代码的嵌入式 sql，进行表的创建，结果的输出。

对于 medium_relation 的数据集求相似度的任务，有如下公式：

|A| + |B| = |A| ^|B| + |A| U |B|

通过先求交集，再通过数学公式求并集，最后交集除以并集，可减少计算量。SQL 主代码如下：

```java
package cn.edu.shu;

import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.table.api.EnvironmentSettings;
import org.apache.flink.table.api.TableEnvironment;
import org.apache.flink.table.api.bridge.java.StreamTableEnvironment;

public class medium_result {
    public static void main(String[] args) throws Exception {
        // StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
        // StreamTableEnvironment tableEnv = StreamTableEnvironment.create(env);

        // 使用批处理模式
        EnvironmentSettings settings = EnvironmentSettings.newInstance().inBatchMode().build();
        TableEnvironment tableEnv = TableEnvironment.create(settings);

        // env.setParallelism(1);

        String sourceDDL = "CREATE TABLE medium_relation ("
            + "referrer INT,"
            + "referree INT"
            + ") WITH ("
            + " 'connector' = 'filesystem', "
            + " 'path' = 'file:///home/ubuntu/medium_relation', "
            + " 'format' = 'csv', "
            + " 'csv.field-delimiter' = ' ', "
            + " 'csv.ignore-parse-errors' = 'true' "
            + ")";
        tableEnv.executeSql(sourceDDL);

        // 修改输出表为文件系统
        tableEnv.executeSql("CREATE TABLE SinkTable ("
            + "web1 INT, "
            + "web2 INT, "
            + "similarity DOUBLE"
            + ") WITH ("
            + " 'connector' = 'filesystem',"
            + " 'path' = 'file:///home/ubuntu/tableapp/output/medium_result',"
            + " 'format' = 'csv',"
            + " 'csv.field-delimiter' = ',',"
            + " 'sink.rolling-policy.file-size' = '256MB',"
            + " 'sink.rolling-policy.rollover-interval' = '30 min'"
            + ")");
```

```java
        // 创建临时视图 ref_count
        tableEnv.executeSql("CREATE TEMPORARY VIEW tmp_ref_count AS "
            + "SELECT referrer, COUNT(DISTINCT referree) AS web_count "
            + "FROM medium_relation "
            + "GROUP BY referrer");

        // 创建临时视图 common
        tableEnv.executeSql("CREATE TEMPORARY VIEW tmp_common AS "
            + "SELECT a.referrer AS web1, b.referrer AS web2, COUNT(*) AS com_cnt "
            + "FROM medium_relation a "
            + "JOIN medium_relation b ON a.referree = b.referree "
            + "WHERE a.referrer < b.referrer "
            + "GROUP BY a.referrer, b.referrer");

        // 计算相似度并插入结果表
        tableEnv.executeSql("INSERT INTO SinkTable "
            + "SELECT "
            + "    common.web1 AS web1, "
            + "    common.web2 AS web2, "
            + "    CASE "
            + "        WHEN r1.web_count + r2.web_count - common.com_cnt > 0 "
            + "        THEN common.com_cnt * 1.0 / (r1.web_count + r2.web_count - common.com_cnt) "
            + "        ELSE 0 "
            + "    END AS similarity "
            + "FROM tmp_common common "
            + "JOIN tmp_ref_count r1 ON common.web1 = r1.referrer "
            + "JOIN tmp_ref_count r2 ON common.web2 = r2.referrer");
    }
}
```

由于输出数据量共六千四百多万行，通过 webui 输出结果容易卡死崩溃，这里将结果输出到了本地的~/tableapp/output/medium_result 目录下，格式为 csv，列属性为网站 a,网站 b,相似度。

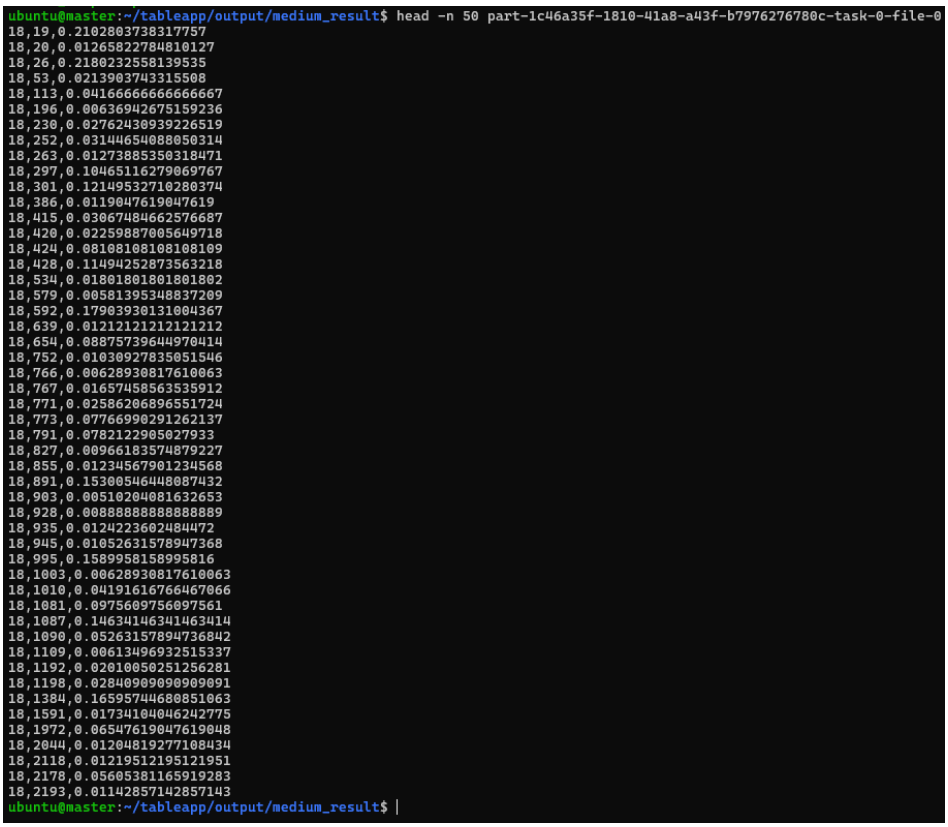
12. 保存好 java 作业代码后，由于已经将 maven 的 bin 目录添加到了换将变量，所以可直接在含有 pom.xml 的目录下，输入如下命令对项目进行打包编译：

```
mvn package
```

13. 然后提交 flink 作业

```
flink run
```

如下图，使用如下命令：
Head -n 50［文件名］
对于结果文件，输出了前 50 行结果



为了确认结果文件的数据量的大小，改名后，使用如下命令查看结果文件的行数：
wc -l medium_output
文件行数，共 64195979 行，数量正确，实验结果输出无误，实验成功。