Flink + kafka + FlinkSql 计算 10秒滚动窗口内用户点击次数，之后自定义 sink To mysql

Flink+kafka 流数据 使用FlinkSql 计算 10秒滚动窗口内 用户点击次数，之后自定义 sink To mysql。

Flink版本为1.6.1

代码如下：

FlinkSqlWindowUserPv.java

```
import java.sql.Timestamp;
import java.util.Properties;

import org.apache.flink.api.common.functions.MapFunction;
import org.apache.flink.api.common.serialization.SimpleStringSchema;
import org.apache.flink.api.common.typeinfo.Types;
import org.apache.flink.api.java.tuple.Tuple3;
import org.apache.flink.api.java.tuple.Tuple5;
import org.apache.flink.streaming.api.datastream.DataStream;
import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.streaming.connectors.kafka.FlinkKafkaConsumer010;
import org.apache.flink.table.api.Table;
import org.apache.flink.table.api.TableConfig;
import org.apache.flink.table.api.java.StreamTableEnvironment;

import pojo.UserPvEntity;


public class FlinkSqlWindowUserPv{

    public static void main(String[] args) throws Exception {

        StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecut

        env.setParallelism(8);

        TableConfig tc = new TableConfig();

        StreamTableEnvironment tableEnv = new StreamTableEnvironment(env, tc)
```

```
        Properties properties = new Properties();                    properties.put("
        properties.put("zookeeper.connect", "127.0.0.1:2181");
        properties.put("key.deserializer", "org.apache.kafka.common.serializa
        properties.put("value.deserializer", "org.apache.kafka.common.serial
        properties.put("group.id", "test6");

        FlinkKafkaConsumer010<String> myConsumer = new FlinkKafkaConsumer010<
                properties);

        DataStream<String> stream = env.addSource(myConsumer);

        DataStream<Tuple5<String, String, String, String, Long>> map = strear

                private static final long serialVersionUID = 1471936326697828

                        @Override
                        public Tuple5<String, String, String, String,Long> ma
                                String[] split = value.split(" ");
                                return new Tuple5<String, String, String, Str
                        }
                });

        map.print(); //打印流数据


        //注册为user表
        tableEnv.registerDataStream("Users", map, "userId,itemId,categoryId,b

        //执行sql查询     滚动窗口 10秒     计算10秒窗口内用户点击次数
        Table sqlQuery = tableEnv.sqlQuery("SELECT TUMBLE_END(proctime, INTER
                        + "userId,count(*) as pvcount "
                        + "FROM Users "
                        + "GROUP BY TUMBLE(proctime, INTERVAL '10' SECOND), u


        //Table 转化为 DataStream
        DataStream<Tuple3<Timestamp, String, Long>> appendStream = tableEnv.t

        appendStream.print();


        //sink to mysql
        appendStream.map(new MapFunction<Tuple3<Timestamp,String,Long>, UserP

                        private static final long serialVersionUID = -477096

                        @Override
                        public UserPvEntity map(Tuple3<Timestamp, String, Lon
```

```
                                                                return ne
                        }
                }).addSink(new SinkUserPvToMySQL2());

        env.execute("userPv from Kafka");

    }

}
```

SinkUserPvToMySQL2.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

import org.apache.flink.configuration.Configuration;
import org.apache.flink.streaming.api.functions.sink.RichSinkFunction;

import pojo.UserPvEntity;

public class SinkUserPvToMySQL2 extends RichSinkFunction<UserPvEntity> {

        private static final long serialVersionUID = -4431754303719194O7L;
        PreparedStatement ps;
    private Connection connection;

    /**
     * open() 方法中建立连接，这样不用每次 invoke 的时候都要建立连接和释放连接
     *
     * @param parameters
     * @throws Exception
     */
    @Override
    public void open(Configuration parameters) throws Exception {
        super.open(parameters);
        connection = getConnection();
        String sql = "replace into t_user_pv(pvtime,userId, pvcount) values(
        ps = this.connection.prepareStatement(sql);
    }

    @Override
    public void close() throws Exception {
        super.close();
        //关闭连接和释放资源
        if (connection != null) {
            connection.close();
```

```
        }
                if (ps != null) {
            ps.close();
        }
    }

    /**
     * 每条数据的插入都要调用一次 invoke() 方法
     *
     * @param value
     * @param context
     * @throws Exception
     */
    @Override
    public void invoke(UserPvEntity userPvEntity, Context context) throws Exc
        //组装数据，执行插入操作
        ps.setLong(1, userPvEntity.getTime());
        ps.setString(2, userPvEntity.getUserId());
        ps.setLong(3, userPvEntity.getPvcount());

        ps.executeUpdate();
    }

    private static Connection getConnection() {
        Connection con = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/my
        } catch (Exception e) {
            System.out.println("-----------mysql get connection has exceptio
        }
        return con;
    }
}
```

结果展示：

```
[root@flink-master bin]# ./kafka-console-producer.sh --broker-list [redacted]:9092 --topic myItems_topic5
>543462 1715 1464116 cart 1511658000
>543462 1715 1464116 cart 1511658000
>543462 1715 1464116 cart 1511658000
>543462 1715 1464116 cart 1511658000
>543462 1715 1464116 cart 1511658000
>543462 1715 1464116 cart 1511658000
>543462 1715 1464116 cart 1511658000
>543462 1715 1464116 cart 1511658000
>543462 1715 1464116 cart 1511658000
>543462 1715 1464116 cart 1511658000
>543462 1715 1464116 cart 1511658000
```

```
8> (543462,1715,1464116,cart,1511658000000)
3> (2018-12-10 02:31:00.0,543462,1)
1> (543462,1715,1464116,cart,1511658000000)
3> (2018-12-10 02:31:10.0,543462,1)
8> (543462,1715,1464116,cart,1511658000000)
1> (543462,1715,1464116,cart,1511658000000)
3> (2018-12-10 02:31:20.0,543462,2)
8> (543462,1715,1464116,cart,1511658000000)
3> (2018-12-10 02:31:30.0,543462,1)
```