Flink入门-2

我又来了，今天玩点基础的吧。看看flink的一些基础功能。还接上用上次的例子，如果没看的可以看一下【Flink入门-1】。

1.map函数。例如下面这个。很常用就是Lambda表达式。对数组[1,2,3]分别带入函数i\*i。计算结果。

env.fromElements(1, 2, 3)

.map(i -> i\*i)

.print();

结果：

[test@TOGGSERAP01 flink-1.6.0]$ ./bin/flink run flink-test-1.0-SNAPSHOT.jar

Starting execution of program

1

4

9

2. flatMap函数，官方的例子不太好懂。修改一下如下所示，理解起来就简单多了。flatMap和map的不同很简单。map就是执行对应的操作。flatMap就是将map的结果放到out集合中。

// collector type must be declared

input.flatMap((Integer number, Collector<String> out) -> {

StringBuilder builder = new StringBuilder();

for(int i = 0; i < number; i++) {

builder.append("a");

}

out.collect(builder.toString());

})

.returns(Types.STRING)

.print();

结果：

[test@TOGGSERAP01 flink-1.6.0]$ ./bin/flink run flink-test-1.0-SNAPSHOT.jar

Starting execution of program

a

aa

aaa

3.用类替代，这个就是用类来实现Lambda表达式，但是官方文档上有一个坑，MapFunction是一个接口，直接extends会报错。这块我也被坑了一下。

env.fromElements(1, 2, 3)

.map(new MyTuple2Mapper())

.print();

public static class MyTuple2Mapper implements MapFunction<Integer, Tuple2<Integer, Integer>> {

@Override

public Tuple2<Integer, Integer> map(Integer i) {

return Tuple2.of(i, i);

}

}

结果：

(1,1)

(2,2)

(3,3)

(1,1)

(2,2)

(3,3)

经过了上面这些枯燥的东西，下面看看Word Count这个例子吧。

DataSet<Tuple2<String, Integer>> counts =

text.flatMap(newTokenizer())

.groupBy(0)

.sum(1);

public static class Tokenizer implements FlatMapFunction<String, Tuple2<String, Integer>> {

@Override

public void flatMap(String value, Collector<Tuple2<String, Integer>> out) {

String[] tokens = value.toLowerCase().split("\\W+");

for (String token : tokens) {

if (token.length() > 0) {

out.collect(new Tuple2<String, Integer>(token, 1));

}

}

}

}

这个逻辑，可以分成flatMap-> groupBy-> sum三步，flatMap实现的目的是将文本分成（单词，1）的结构。groupBy（0）就是按逐个单词来分组，sum则为加各。最终得到的结果就是下面这样。

1 1

13 1

5d002 1

740 1

about 1

真得很方便。比之前写hadoop的mr要方便很多。但是原理其实是一样的。后面我会继续学习Batch的api。然后再进行DataStream的学习。还有Table API。这几个学完基本就差不多了。就到这里，休息一下。