

## 华东师范大学数据学院上机实践报告

课程名称：分布式模型与编程 年级：2018 上机实践成绩：  
指导教师：徐辰 姓名：孙秋实  
上机实践名称：Flink 编程 学号：10185501402 上机实践日期：2021/5/13  
上机实践编号：Lab14 组号：Group5 上机实践时间：

### Part 1

#### 实验目的

- (1) 学习编写简单的基于 DataStream API 的 Flink 程序
  - (2) 掌握在 IDEA 中调试 Flink 相关程序，以及在单机伪分布式、分布式模式下提交运行 Flink 相关程序的方法。
- 

### Part 2

#### 实验任务

- (1) 完成 WordCount 示例程序的编写。
  - (2) 在单机伪分布式和分布式模式下运行 WordCount 示例程序
- 

### Part 3

#### 使用环境

- (1) 操作系统：Ubuntu 18.04
  - (2) JDK 版本：1.8
  - (3) Hadoop 版本：2.10.1
  - (4) Flink 版本：1.12.1
  - (5) Scala 版本：2.11.12
- 

### Part 4

#### 实验过程

### Section 1

#### 编写 Flink 应用程序

新建一个 Maven 项目并且添加依赖

```

<dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-streaming-java_2.11</artifactId>
    <version>1.12.1</version>
    <scope>compile</scope>
</dependency>

```

图 1: 配置 pom.xml 文件

```

package cn.edu.ecnu.flink.example.java.wordcount;

import org.apache.flink.api.common.functions.FlatMapFunction;
import org.apache.flink.api.common.functions.MapFunction;
import org.apache.flink.api.java.tuple.Tuple2;
import org.apache.flink.streaming.api.datastream.DataStream;
import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.util.Collector;

public class WordCount {
    public static void main(String[] args) throws Exception {
        run(args);
    }

    public static void run(String[] args) throws Exception {
        /* 步骤1：创建StreamExecutionEnvironment对象 */
        StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();

        /* 步骤2：按应用逻辑使用操作算子编写DAG，操作算子包括数据源、转换、数据池等 */
        // 从指定的主机名和端口号接收数据，创建名为lines的数据流
        DataStream<String> lines = env.socketTextStream(args[0], Integer.parseInt(args[1]));
        // 将lines中的每一个文本行按空格分割成单个单词
        DataStream<String> words =
            lines.flatMap(
                new FlatMapFunction<String, String>() {
                    @Override
                    public void flatMap(String value, Collector<String> out) throws Exception {
                        for (String word : value.split(" ")) {
                            out.collect(word);
                        }
                    }
                }
            );
    }
}

```

图 2: 编写 WordCount 程序

(这里文件路径忘记换，显示的是 Spark，但实验是 Flink 的实验)

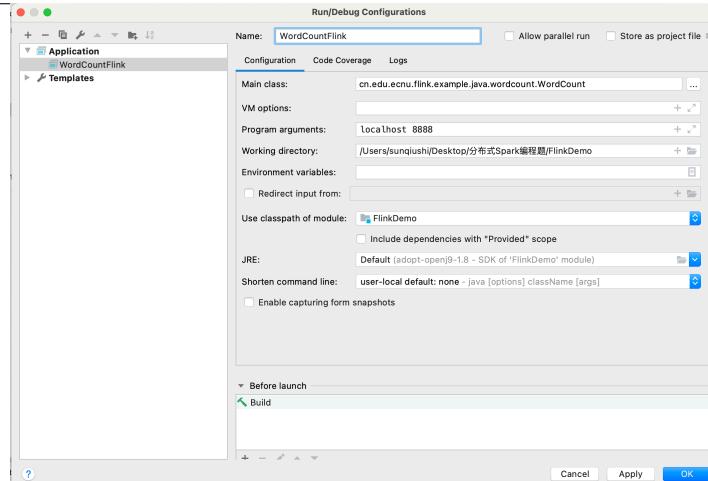


图 3: 调试 Flink 应用程序

### Section 3

#### 运行 Flink 应用程序

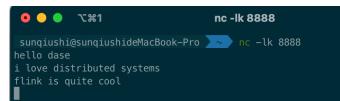


图 4: 启动 Netcat 监听 8888 端口，并输入

可以看到 idea 的命令行做出了反应，显示了词频统计的结果

我们可以在 idea 的命令行查看输出结果

```
3> (hello,1)
7> (dase,1)
6> (love,1)
5> (distributed,1)
2> (is,1)
8> (systems,1)
7> (flink,1)
8> (cool,1)
5> (quite,1)
```

图 5: 在 idea 命令行查看输出结果

我们现在中断 Netcat 的监听，中断 Netcat 的监听后可以看到 idea 的命令行中也给出了断开连接的提示

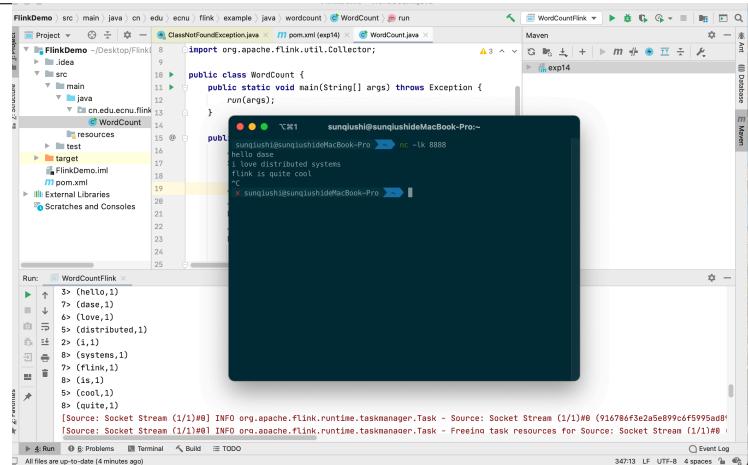


图 6: 中断 Netcat 监听

制作 WordCount 的 jar 包

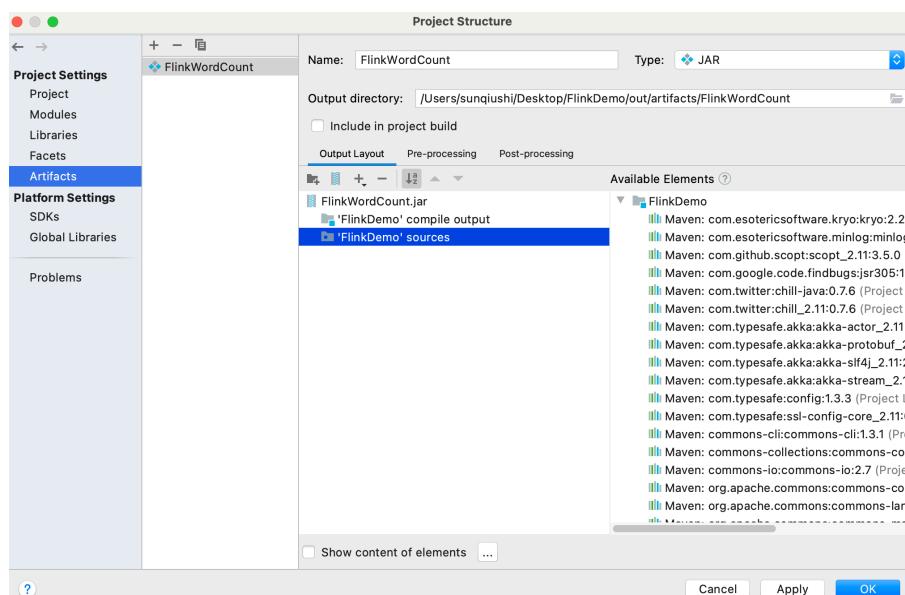


图 7: 制作 WordCount 的 Jar 包



图 8: 制作 WordCount 的 Jar 包 Cont'd

在接下来的伪分布式和集中式部署中我们需要用到这个 jar 包

#### Section 4

##### 单机伪分布式部署

实验开始前先把刚刚制作的 jar 包从本地拷贝到云主机的 dase-local

## 华东师范大学数据科学与工程学院学生实验报告

```
sunqilushi@sunqiuhideMacBook-Pro ~ % scp /Users/sunqilushi/Desktop/lab14-Flink  
编程/FlinkWordCount.jar dase-local@10.24.21.125:/softwares/flink-1.12.1/myApp  
FlinkWordCount.jar          100% 4887   336.3KB/s  00:00  
sunqilushi@sunqiuhideMacBook-Pro ~ %  
● ● ● ℗%1 dase-local@10-24-21-125: ~/softwares/flink-1.12.1/myApp  
dase-local@10-24-21-125:~/softwares/flink-1.12.1$ ls  
LICENSE README.txt conf lib log plugins  
NOTICE bin examples licenses opt  
dase-local@10-24-21-125:~/softwares/flink-1.12.1$ mkdir myApp  
dase-local@10-24-21-125:~/softwares/flink-1.12.1$ cd myApp/  
dase-local@10-24-21-125:~/softwares/flink-1.12.1/myApp$ ls  
FlinkWordCount.jar  
dase-local@10-24-21-125:~/softwares/flink-1.12.1/myApp$
```

图 9: 从本地拷贝文件到云主机

伪分布式模式下启动 flink (启动 flink 集群稍微要一点时间)

```
dase-local@10-24-21-125:~/softwares/flink-1.12.1/bin$ ./start-cluster.sh  
Starting cluster.  
Starting standalone session daemon on host 10-24-21-125.  
Starting taskexecutor daemon on host 10-24-21-125.  
dase-local@10-24-21-125:~/softwares/flink-1.12.1/bin$
```

图 10: 启动 flink 集群

启动后查看进程状态，确认运行正常

```
dase-local@10-24-21-125:~/softwares/flink-1.12.1/bin$ jps  
6081 Jps  
5705 StandaloneSessionClusterEntrypoint  
5979 TaskManagerRunner  
dase-local@10-24-21-125:~/softwares/flink-1.12.1/bin$
```

图 11: 使用 jps 查看进程状态

启动 Netcat 开始监听，并在提交任务后输入用于词频统计的文本

```
sunqilushi@sunqiuhideMacBook-Pro ~ % ssh dase-local@10.24.21.125  
Last login: Thu May 13 23:54:07 2021 from 219.228.146.139  
dase-local@10-24-21-125:~$ nc -l 8888  
i love flink  
no one loves flink better than me  
i love data science and engineering
```

图 12: 使用 Netcat 监听

通过 jar 包提交程序

```
● ● ● ℗%1 dase-local@10-24-21-125: ~/softwares  
dase-local@10-24-21-125:~/softwares$ ./softwares/flink-1.12.1/bin/flink run -c  
n.edu.ecnu.flink.example.java.wordcount.WordCount ~/softwares/flink-1.12.1/myApp  
/FlinkWordCount.jar localhost 8888  
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/dase-local/softwares/flink-1.12.1/lib/lo  
g4j-slf4j-impl-2.12.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/home/dase-local/softwares/hadoop-2.10.1/share  
/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.c  
lass]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]  
Job has been submitted with JobID 1a917998b95a639df06a3968336404c3  
|
```

图 13: 提交 WordCount 程序

随后我们可以在这个机器的 flink 目录下的 log 文件夹看到运行结果

```
dase-local@10-24-21-125:~/softwares/flink-1.12.1/log$ ls
flink-dse-local-client-10-24-21-125.log
flink-dse-local-scala-shell-local-10-24-21-70.log
flink-dse-local-scala-shell-remote-10-24-21-70.log
flink-dse-local-standaloneSession-0-10-24-21-125.log
flink-dse-local-standaloneSession-0-10-24-21-125.out
flink-dse-local-standaloneSession-0-10-24-21-70.log
flink-dse-local-standaloneSession-0-10-24-21-70.out
flink-dse-local-taskexecutor-0-10-24-21-125.log
flink-dse-local-taskexecutor-0-10-24-21-125.out
flink-dse-local-taskexecutor-0-10-24-21-70.log
flink-dse-local-taskexecutor-0-10-24-21-70.out
dase-local@10-24-21-125:~/softwares/flink-1.12.1/log$
```

图 14: 查看日志文件

我们打开以节点名命名的.out 文件查看词频统计结果

```
dase-local@10-24-21-125:~/softwares/flink-1.12.1/log$ tail flink-dse-local-taskexecutor-0-10-24-21-125.out
(flink,2)
(better,1)
(than,1)
(me,1)
(i,2)
(love,2)
(data,1)
(science,1)
(and,1)
(engineering,1)
dase-local@10-24-21-125:~/softwares/flink-1.12.1/log$
```

图 15: 打开.out 文件查看词频统计结果

提交程序后，我们可以在 Flink 的 Web UI 中看到运行状况，可以看到有一个 job 正在 running

Job Name	Start Time	Duration	End Time	Tasks	Status
Streaming WordCount	2021-05-14 00:06:41	8m 10s	-	2	RUNNING

图 16: Flink Web UI

结束 Netcat 的监听后可以看到状态由 running 转为了 finished

Job Name	Start Time	Duration	End Time	Tasks	Status
Streaming WordCount	2021-05-14 00:06:41	8m 21s	2021-05-14 00:15:03	2	FINISHED

图 17: 任务结束

然后停止 Flink，结束伪分布式下实验

## 华东师范大学数据科学与工程学院学生实验报告

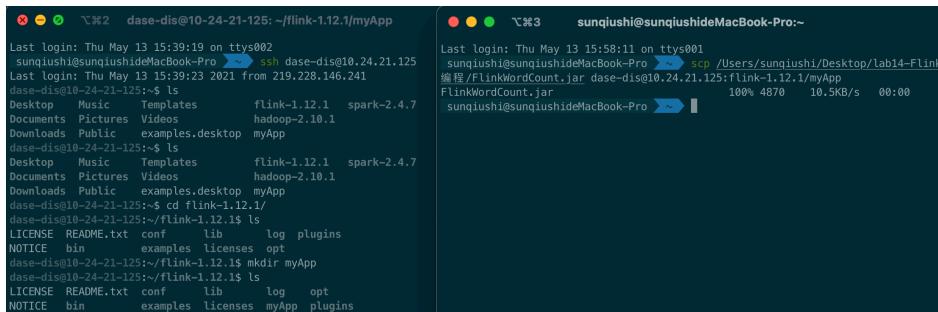
```
dase-dis@10-24-21-125:~/softwares/flink-1.12.1/bin$ stop-cluster.sh
Stopping taskexecutor daemon (pid: 5979) on host 10-24-21-125.
Stopping standalonesession daemon (pid: 5705) on host 10-24-21-125.
dase-dis@10-24-21-125:~/softwares/flink-1.12.1/bin$
```

图 18: 结束实验

### Section 5

#### 分布式部署

实验同样由四台机器完成，分别为一个主节点，两个从节点，一个客户端  
首先需要把我们准备好的 jar 包传输到 Flink 的目录下



The image shows two terminal windows. The left window is on a Mac (MacBook-Pro) with the command: `scp /Users/sunqiuishi/Desktop/lab14-Flink 编程/FlinkWordCount.jar dase-dis@10.24.21.125:flink-1.12.1/myApp`. The right window is on a Linux host (dase-dis@10-24-21-125) with the command: `ls` showing the transferred jar file.

图 19: 拷贝文件至云主机 Flink 目录下

```
dase-dis@10-24-21-144:~/flink-1.12.1/bin$ ./start-cluster.sh
Starting cluster.
Starting standalonesession daemon on host 10-24-21-144.
Starting taskexecutor daemon on host ecnu02.
Starting taskexecutor daemon on host ecnu03.
dase-dis@10-24-21-144:~$ jps
5463 StandaloneSessionClusterEntrypoint
5529 Jps
dase-dis@10-24-21-144:~$
```

图 20: 主节点启动并查看进程状态

主节点启动后，客户端启动 Netcat 服务，再提交之前编写的 WordCount

```
dase-dis@10-24-21-125:~/flink-1.12.1$ cd myApp/
dase-dis@10-24-21-125:~/flink-1.12.1/myApp$ ls
FlinkWordCount.jar
dase-dis@10-24-21-125:~/flink-1.12.1/myApp$ ./flink-1.12.1/bin/flink run -c cn.edu.ecnu.flink.example.java.wordcount.WordCount ~/flink-1.12.1/myApp/FlinkWordCount.jar ecnu04 8888
Job has been submitted with JobID aced87362de81d692af6ced56b2e8c3
```

图 21: 客户端提交 jar 包

随后在监听的端口为程序提供输入数据

```
dase-dis@10-24-21-125:~/flink-1.12.1$ nc -lk 8888
wengsiyang nb
i love dase
i love flink
```

图 22: 通过 Netcat 输入用于词频统计文本

在启动 Netcat 服务的终端中输入数据后，主节点可以访问 Web UI 的方式查看运行状态，以及程序运行在哪个从节点（从 Hosts 条目看）

## 华东师范大学数据科学与工程学院学生实验报告

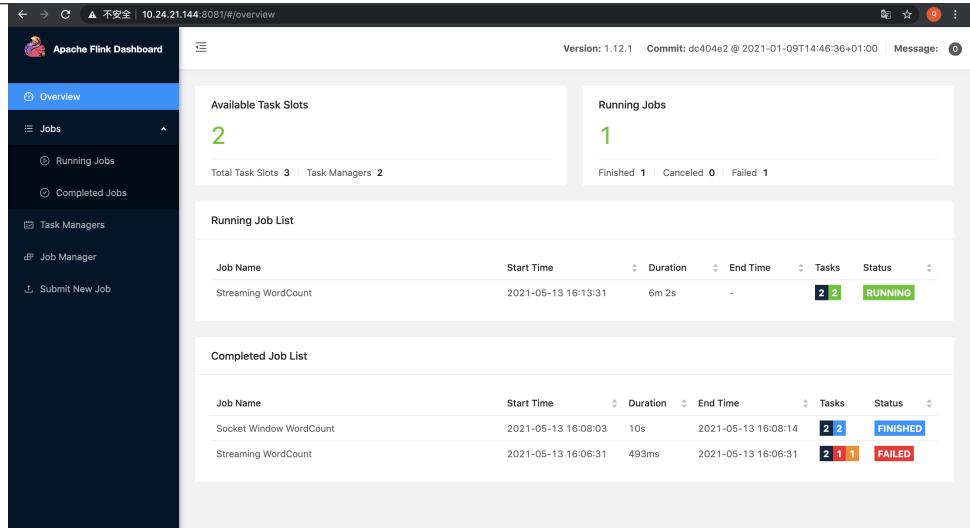


图 23: 主节点查看 Web UI

Remark: 我们可以看到这时候的 Job 状态为 Running，它会一直保持 Running 直到客户端退出监听

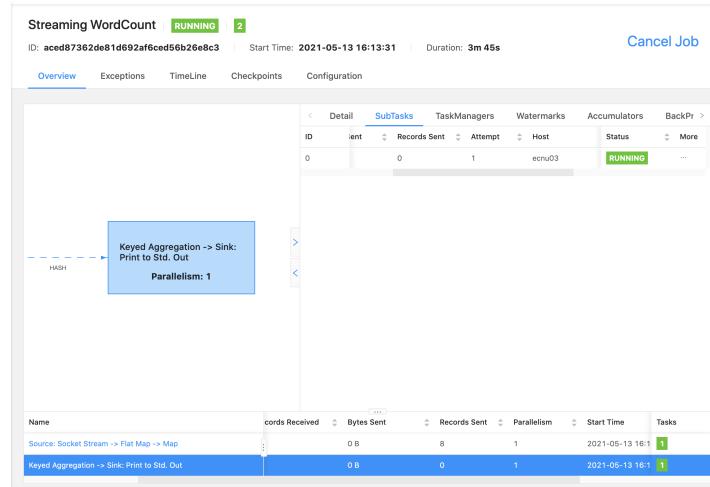


图 24: 主节点查看 Flink 应用程序界面

我们确定是 ecnu03 节点执行了我们的词频统计任务，随后在 ecnu03 上可以查看到输出结果

```
dase-dis@ecnu03:~$ tail flink-1.12.1/log/flink-dase-dis-taskexecutor-0-ecnu03.out
(wengsiyang,1)
(nb,1)
(i,1)
(love,1)
(dase,1)
(i,2)
(love,2)
.flink,1)
```

图 25: 执行任务的从节点查看结果

Remark: 在客户端输入更多数据后，可以看到这个文件内容的更新

```
dase-dis@10-24-21-125:~/flink-1.12.1$ nc -l 8888
wengsiyang nb
i love dase
i love flink
^C
dase-dis@10-24-21-125:~/flink-1.12.1$
```

图 26: 结束 Netcat 监听

通过 Ctrl-C 结束 Netcat 后，可以在主节点的 Web UI 中看到进程状态从 Running 变为 Finished

The screenshot shows the Flink Web UI at the URL [10.24.21.144:8081/#/overview](http://10.24.21.144:8081/#/overview). The left sidebar has sections for Jobs (Running Jobs, Completed Jobs), Task Managers, Job Manager, and Submit New Job. The main area has tabs for Available Task Slots (3 slots), Running Jobs (0), and Completed Job List. The Running Job List table is empty with a 'No Data' message. The Completed Job List table shows three entries:

Job Name	Start Time	Duration	End Time	Tasks	Status
Streaming WordCount	2021-05-13 16:13:31	6m 16s	2021-05-13 16:19:48	2 2	FINISHED
Socket Window WordCount	2021-05-13 16:08:03	10s	2021-05-13 16:08:14	2 2	FINISHED
Streaming WordCount	2021-05-13 16:06:31	493ms	2021-05-13 16:06:31	2 1 1	FAILED

图 27: 客户端从 Netcat 退出后，Web UI 中看到进程状态变化

随后主节点停止 Flink 服务，结束分布式模式下的实验

```
[dase-dis@10-24-21-144:~$ ~/flink-1.12.1/bin/stop-cluster.sh
Stopping taskexecutor daemon (pid: 6356) on host ecnu02.
Stopping taskexecutor daemon (pid: 5280) on host ecnu03.
Stopping standalonesession daemon (pid: 6305) on host 10-24-21-144.
[dase-dis@10-24-21-144:~$ jps
6851 Jps
```

图 28: 结束实验

## Part 6

思考题

### Section 1

如何查看一个 Flink 流计算应用程序启动了多少个任务，以及每个任务的并行度是多少？请结合 Web UI 来说明

根据参考理论书第 172 页图 10.13，其实任务对应的就是线程，我们在这里考虑算子的并行度。

根据 Flink UI，前面提交词频统计任务一共有两个算子，并行度均为 1，所以各开启了一个任务（Task），也即线程/任务数为 2。

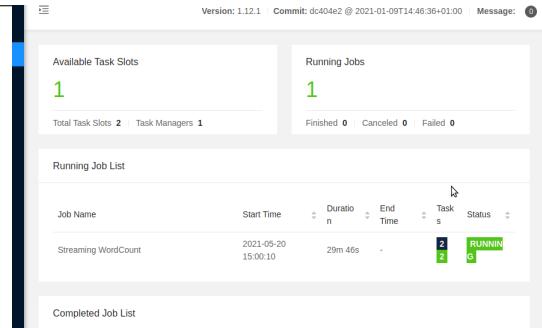


图 29: 思考题 1-1



图 30: 思考题 1-2

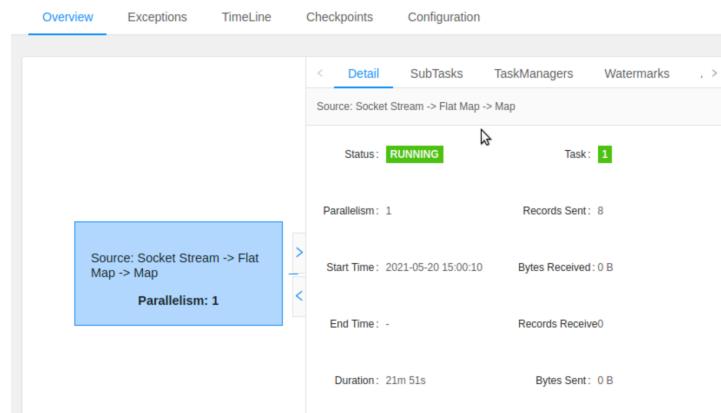


图 31: 思考题 1-3

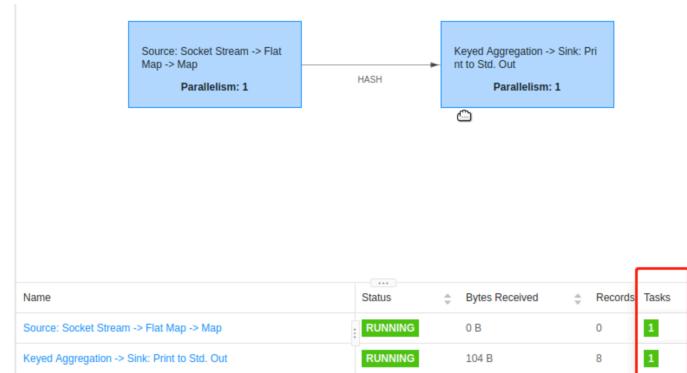


图 32: 思考题 1-4

## Section 2

一个 Flink 应用程序中的算子会发生合并吗？哪些算子会合并？

一个 Flink 应用程序中的算子会发生合并，如图 30 所示，在上述任务的 Flink UI 中可以看到相关信息。；

- Socket Stream、FlatMap 和 Map 被合并为 1 个算子
- Keyed Aggregation 与 print 被合并为 1 个算子

Remark: Flink 使用 Chaining 机制将有一对一数据传输关系的算子合并为一个大算子，避免它们被分配到不同的 Task Manager。