

Chapter 4. Setting Up a Development Environment for Apache Flink

(启动一个 flink 开发环境)

1. 前言

本章中，学习如何 设置环境开发、运行、调试 Flink 应用程序。 讨论程序所需的环境，并获取本书的代码示例。 使用这些示例来展示如何在 IDE 中执行和调试 Flink 应用程序。 最后，会展示了如何引导 Flink Maven 项目

2. Required Software

需要的软件

2.1. 开发和运行的环境

- *unix 环境有最丰富的软件支持，也可以使用 Linux (WSL)，Cygwin 或 Linux 虚拟机的 windows 子系统

- *可以使用 java 或者 scala 开发，java 需要 1.8 或更高

- *maven

- *IntelliJ IDEA

3. Run and Debug Flink Applications in an IDE

在 ide 中调试和运行 flink 程序

3.1. Import the Book' s Examples in an IDE （导入本书的例子到 IDE）

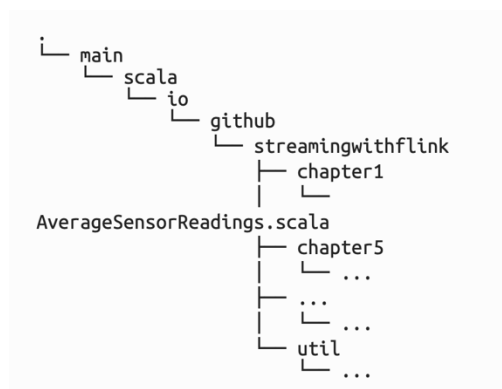
*git clone <https://github.com/streaming-with-flink/examples-scala>

*也可以使用命令下载 zip 压缩包

> wget <https://github.com/streaming-with-flink/examples-scala/archive/master.zip>

> unzip master.zip

*项目结构如图



*使用 idea 打开项目，这里就不展开了

Run Flink Applications in an IDE （在 ide 中运行 flink 程序）

a 程序运行说明

找到 AverageSensorReadings 这个类，程序逻辑 是读取多个传感器的温度，并从华氏温度转换为摄氏温度，并每秒计算每个传感器的平均温度，程序的 source, sink, operators 都在 main 方法中，启动动程序后将在控制台输出结果。

控制台的输出结果如图

```
2>
SensorReading(sensor_31,1515014051000,23.92465618384873
2)
4>
SensorReading(sensor_32,1515014051000,4.118569049862492
)
1>
SensorReading(sensor_38,1515014051000,14.78183542024247
1)
3>
SensorReading(sensor_34,1515014051000,23.87143325225058
3)
```

控制台的输出结果

b 程序幕后执行逻辑

在执行 StreamExecutionEnvironment.execute() 方法后 main() 方法将组合的 dataflow 提交到远程 Jobmanage(master)，Jobmanager 将分发 task 给不同的 TaskManager(worker)，Jobmanager 和 TaskManager 是运行在不同机器上的 jvm 进程

IDE 调试的模式：在调用 `execute()`后，在相同的 jvm 中启动 `jobmanager` 和 `taskmanager`(默认 slot 和 cpu 线程数一样多)，这时整个 flink 程序是多线程的并在同一 jvm 中，这种模式适合在 ide 中启动 flink

3.2. Debug Flink Applications in an IDE

- *在 ide 调试模式下，可以像普通程序一样打断点调试
- *默认启动的线程数和 cpu 线程数相同，是多线程的程序
- *与通过将 Flink 程序发送到远程 JobManager 来执行 Flink 程序相反，此模式为单个 JVM 中执行，类的加载无法正确调试
- *虽然是在单个 jvm 中运行，records 是要进行序列化来进行跨线程通信

4. Bootstrap a Flink Maven Project

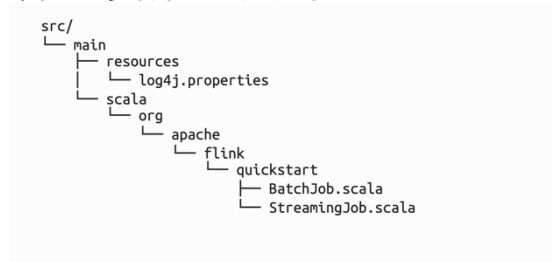
(引导 flink maven 项目)

主要写了如何使用 maven 命令创建一个 flink 项目

4.1. 使用如下 maven 命令创建一个名为 quickstart 的项目

```
mvn archetype:generate \
  -DarchetypeGroupId=org.apache.flink \
  -DarchetypeArtifactId=flink-quickstart-scala \
  -DarchetypeVersion=1.7.1 \
  -DgroupId=org.apache.flink.quickstart \
  -DartifactId=flink-scala-project \
  -Dversion=0.1 \
  -Dpackage=org.apache.flink.quickstart \
  -DinteractiveMode=false
```

项目的结构层次如图



最后使用如下命令在 target 目录生成 jar 包：mvn clean package -Pbuild-jar

5. Summary

本章讲了创建一个 flink 项目，5、6、7、8 主要讲解 api 和 operator、stateful functions、source and sink connectors