# 实验八 **Flink初级编程实践**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **地 点：** | B10楼 | 303房； | **实验台号：** |  |
| **实验日期与时间：** | 2023/6/1 | | **评 分：** |  |
| **预习检查纪录：** |  | | **实验教师：** | 焦青松 |

1. **实验目的**

**（1）.通过实验掌握基本的Flink编程方法。**

**(2).掌握用IntelliJ IDEA工具编写Flink程序的方法。**

1. **实验平台**

(1).操作系统:Linux

(2).Flink版本:1.9.1

(3).Intellij IDEA

**实验步骤（包括实验结果/截图）**

1. **思考讨论题或体会或对改进实验的建议**

## 2. Flink 架构中的服务类型

下面是从Flink官网截取的一张架构图：

在Flink运行时涉及到的进程主要有以下两个： ***JobManager****：主要负责调度task，协调checkpoint已经错误恢复等。当客户端将打包好的任务提交到JobManager之后，JobManager就会根据注册的TaskManager资源信息将任务分配给有资源的TaskManager，然后启动运行任务。TaskManger从JobManager获取task信息，然后使用slot资源运行task；***TaskManager**：执行数据流的task，一个task通过设置并行度，可能会有多个subtask。 每个TaskManager都是作为一个独立的JVM进程运行的。他主要负责在独立的线程执行的operator。其中能执行多少个operator取决于每个taskManager指定的slots数量。Task slot是Flink中最小的资源单位。假如一个taskManager有3个slot，他就会给每个slot分配1/3的内存资源，目前slot不会对cpu进行隔离。同一个taskManager中的slot会共享网络资源和心跳信息。  
当然在Flink中并不是一个slot只可以执行一个task，在某些情况下，一个slot中也可能执行多个task，如下：

1. **实验源代码**

**<?xml version="1.0" encoding="UTF-8"?>**

**<project xmlns="http://maven.apache.org/POM/4.0.0"**

**xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"**

**xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">**

**<modelVersion>4.0.0</modelVersion>**

**<groupId>dblab</groupId>**

**<artifactId>FlinkWordCount</artifactId>**

**<version>1.0-SNAPSHOT</version>**

**<dependencies>**

**<dependency>**

**<groupId>org.apache.flink</groupId>**

**<artifactId>flink-java</artifactId>**

**<version>1.9.1</version>**

**</dependency>**

**<dependency>**

**<groupId>org.apache.flink</groupId>**

**<artifactId>flink-streaming-java\_2.11</artifactId>**

**<version>1.9.1</version>**

**</dependency>**

**<dependency>**

**<groupId>org.apache.flink</groupId>**

**<artifactId>flink-clients\_2.11</artifactId>**

**<version>1.9.1</version>**

**</dependency>**

**</dependencies>**

**package cn.edu.xmu;**

**import org.apache.flink.api.java.DataSet;**

**import org.apache.flink.api.java.ExecutionEnvironment;**

**public class WordCountData {**

**public static final String[] WORDS=new String[]{"To be, or not to be,--that is the question:--", "Whether \'tis nobler in the mind to suffer", "The slings and arrows of outrageous fortune", "Or to take arms against a sea of troubles,", "And by opposing end them?--To die,--to sleep,--", "No more; and by a sleep to say we end", "The heartache, and the thousand natural shocks", "That flesh is heir to,--\'tis a consummation", "Devoutly to be wish\'d. To die,--to sleep;--", "To sleep! perchance to dream:--ay, there\'s the rub;", "For in that sleep of death what dreams may come,", "When we have shuffled off this mortal coil,", "Must give us pause: there\'s the respect", "That makes calamity of so long life;", "For who would bear the whips and scorns of time,", "The oppressor\'s wrong, the proud man\'s contumely,", "The pangs of despis\'d love, the law\'s delay,", "The insolence of office, and the spurns", "That patient merit of the unworthy takes,", "When he himself might his quietus make", "With a bare bodkin? who would these fardels bear,", "To grunt and sweat under a weary life,", "But that the dread of something after death,--", "The undiscover\'d country, from whose bourn", "No traveller returns,--puzzles the will,", "And makes us rather bear those ills we have", "Than fly to others that we know not of?", "Thus conscience does make cowards of us all;", "And thus the native hue of resolution", "Is sicklied o\'er with the pale cast of thought;", "And enterprises of great pith and moment,", "With this regard, their currents turn awry,", "And lose the name of action.--Soft you now!", "The fair Ophelia!--Nymph, in thy orisons", "Be all my sins remember\'d."};**

**public WordCountData() {**

**}**

**public static DataSet<String> getDefaultTextLineDataset(ExecutionEnvironment env){**

**return env.fromElements(WORDS);**

**}**

**}**