**作业4**

**Spark环境搭建及程序编译运行**

**一、在Linux系统中安装Spark**

假设之前在Linux系统中已经按照伪分布式模式在机器上安装了Hadoop，请在此基础上，按照Local模式安装Spark，并做相关配置，使得Spark可以顺利访问HDFS。

**二、在spark-shell中运行代码**

请使用Local模式启动进入spark-shell环境，并书写几条Scala语句进行测试，然后退出spark-shell环境。（也可以使用 Python）

**三、编写Spark独立应用程序**

请编写一个简单的Scala代码文件（代码中需要包含println语句），然后，分别使用Maven和sbt工具对程序进行编译打包，最后，使用spark-submit命令运行程序。

（也可以使用Python）

**四、实验报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验报告 | | | | |
| 题目 | Spark环境搭建及程序编译运行 | 姓名 | 邓语苏 | 日期 2023/10/11 |
| 实验环境：Hadoop 3.3.6、Java jdk1.8.0\_381、Spark 3.5.0 | | | | |
| 实验内容与完成情况：  **一、在Linux系统中安装Spark**  假设之前在Linux系统中已经按照伪分布式模式在机器上安装了Hadoop，请在此基础上，按照Local模式安装Spark，并做相关配置，使得Spark可以顺利访问HDFS。  将spark安装包放在~/Downloads目录下   1. sudo tar -zxf ~/Downloads/spark-3.5.0-bin-without-hadoop.tgz -C /usr/local/ 2. cd /usr/local 3. sudo mv ./spark-3.5.0-bin-without-hadoop/ ./spark 4. sudo chown -R hadoop:hadoop ./spark   安装后，修改配置文件spark-env.sh   1. cd /usr/local/spark 2. cp ./conf/spark-env.sh.template ./conf/spark-env.sh 3. vim ./conf/spark-env.sh   编辑spark-env.sh⽂件，在第⼀⾏添加以下配置信息   1. export SPARK\_DIST\_CLASSPATH=$(/usr/local/hadoop/bin/hadoop classpath)   配置完成后就可以直接使⽤，不需要像Hadoop运⾏启动命令。通过运⾏Spark⾃带的示例，验证Spark是否安装成功   1. cd /usr/local/spark 2. bin/run-example SparkPi     **二、在spark-shell中运行代码**  请使用Local模式启动进入spark-shell环境，并书写几条Scala语句进行测试，然后退出spark-shell环境。（也可以使用 Python）  （一）启动Spark Shell（使⽤Python语⾔）  启动spark-shell后，会⾃动创建名为sc的SparkContext对象和名为spark的SparkSession对象,如图：    加载text⽂件   1. textFile = spark.read.text("file:///usr/local/spark/README.md”)   加载HDFS⽂件和本地⽂件区别是添加前缀(hdfs://和file:///)进⾏标识。  （二）简单RDD操作  获取该DataFrame的行数和第一行    现在让我们将此 DataFrame 转换为⼀个新的 DataFrame。我们调⽤filter返回⼀个新的DataFrame，其中包含⽂件中⾏的⼦集   1. linesWithSpark = textFile.filter(textFile.value.contains("Spark"))   统计新DataFrame的行数    将转换和行动链接在一起    找到单词最多的行    在这⾥，我们使⽤explode中的函数select，将⾏数据集转换为单词数据集，然后将⽂件中的每个单词计数合并为 2 列的数据帧：“单词”和“计数” groupBy。count要在 shell 中收集字数统计，我们可以调⽤collect：    Spark 还⽀持将数据集拉⼊集群范围的内存缓存中。当重复访问数据时（例如查询⼩型“热”数据集或运⾏ PageRank 等迭代算法时），这⾮常有⽤。作为⼀个简单的示例，让我们将linesWithSpark数据集标记为要缓存：    **三、编写Spark独立应用程序**  请编写一个简单的Scala代码文件（代码中需要包含println语句），然后，分别使用Maven和sbt工具对程序进行编译打包，最后，使用spark-submit命令运行程序。  （也可以使用Python）      在~/Downloads目录下新建SimpleApp.py文件  使用脚本 | | | | |
| 出现的问题： | | | | |
| 解决方案（列出遇到的问题和解决办法，列出没有解决的问题）： | | | | |

（备注：相关实验过程可以截图放入实验报告）