实验六：流计算及 Spark Streaming 编程

实验基本信息：

时间： ： 2024 年 10 月 12 日

实验类型： □验证性 □设计性 ☑综合性

班级： 计算机科学与技术专业（中外合作）4 班 班级代码:

理论老师： 邱开金 实验指导： 邱开金

实验报告提交说明：

本次实验需要撰写实验报告，实验报告填写时以完成实验任务为目的，先简要回答完成实验任务需要的步骤或需要执行的命令代码，再配以结果截图予以说明，图片数量不宜过多，能说明问题即可。最后配上总结，并提交到教师指定位置。

实验目的：

1. 理解流计算的概念。
2. 掌握从文件流和套接字流中提取数据，并处理。
3. 掌握 Flume 数据源的处理。

实验任务：

（作答要求：在每一个问题后写上所需要的命令，如果命令已经能够清楚回答问题，则不需要抓图。在所有命令执行完毕后，在终端输入 history 命令，将该命令执行结果抓图（抓图应能看清命令以及学生所用的电脑的当前时间或当前桌面背景等能区分出是自己做的证据信息）附在最后）

1. 文件流(DStream)

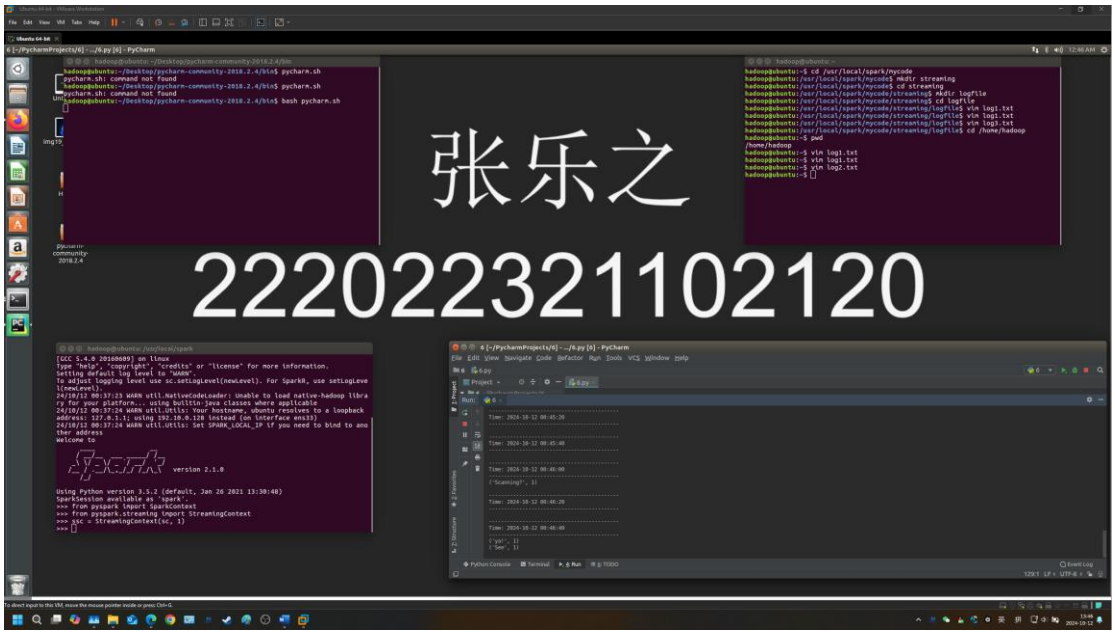
1）编写程序监控/home/hadoop/目录下文件内容的变化，并统计新增的文件中包含的词频输出。

2）注意实验指导书要求的路径与在线教程的差异，应能正确处理径变化。

（需粘贴代码）

```
from operator import add
from pyspark import SparkContext, SparkConf
from pyspark.streaming import StreamingContext
conf = SparkConf()
conf.setAppName('TestDStream')
conf.setMaster('local[2]')
sc = SparkContext(conf = conf)
ssc = StreamingContext(sc, 20)
lines = ssc.textFileStream('file:///home/hadoop/')
```

```
words = lines.flatMap(lambda line: line.split(' '))
wordCounts = words.map(lambda x : (x,1)).reduceByKey(add)
wordCounts.pprint()
ssc.start()
ssc.awaitTermination()
```



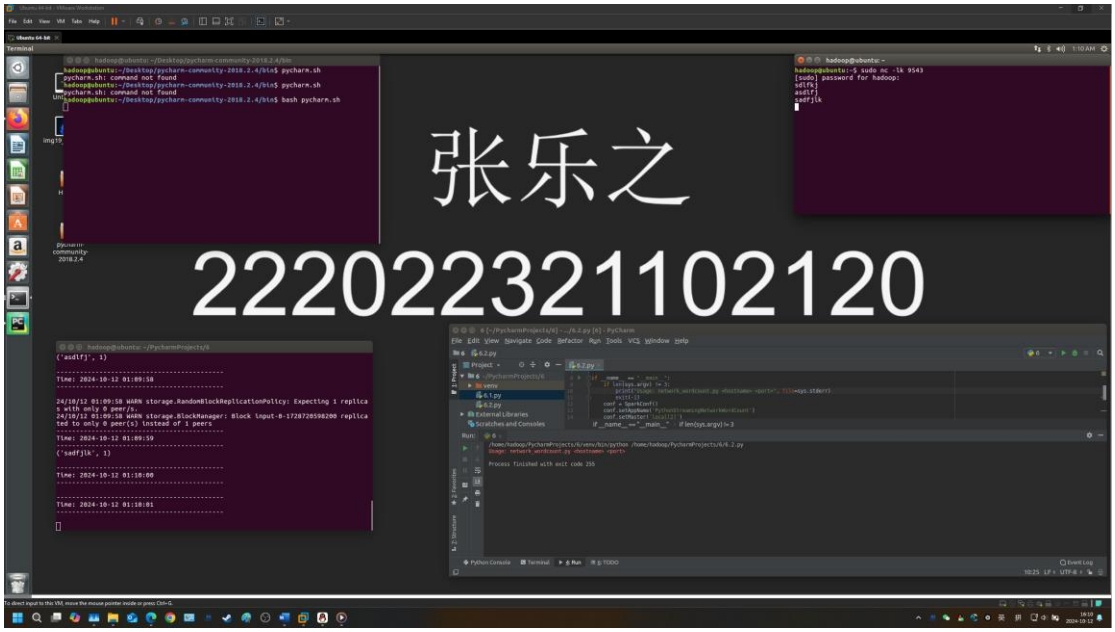
2. 套接字流(DStream)

1）编写程序监控 9543 号端口的数据流，并统计流中的词频并输出。（注意端口号与教程的差异）

```
from __future__ import print_function
from pyspark import SparkConf
import sys

from pyspark import SparkContext
from pyspark.streaming import StreamingContext

if __name__ == "__main__":
        if len(sys.argv) != 3:
                print("Usage: network_wordcount.py <hostname> <port>", file=sys.stderr)
                exit(-1)
        conf = SparkConf()
        conf.setAppName('PythonStreamingNetworkWordCount')
        conf.setMaster('local[2]')
        sc = SparkContext(conf = conf)
        ssc = StreamingContext(sc, 1)

        lines = ssc.socketTextStream(sys.argv[1], int(sys.argv[2]))
        counts = lines.flatMap(lambda line: line.split(" "))\
                                .map(lambda word: (word, 1))\
```

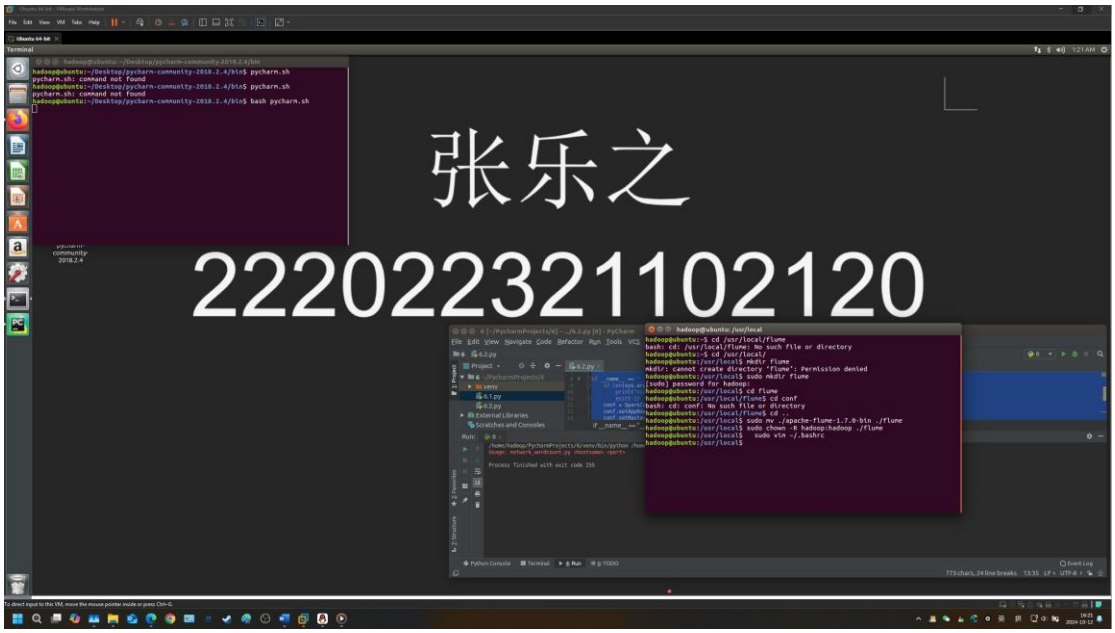
```
                .reduceByKey(lambda a, b: a+b)
    counts.pprint()

    ssc.start()
    ssc.awaitTermination()
```

2）需粘贴代码。



3．Flume 作为 DStream 数据源

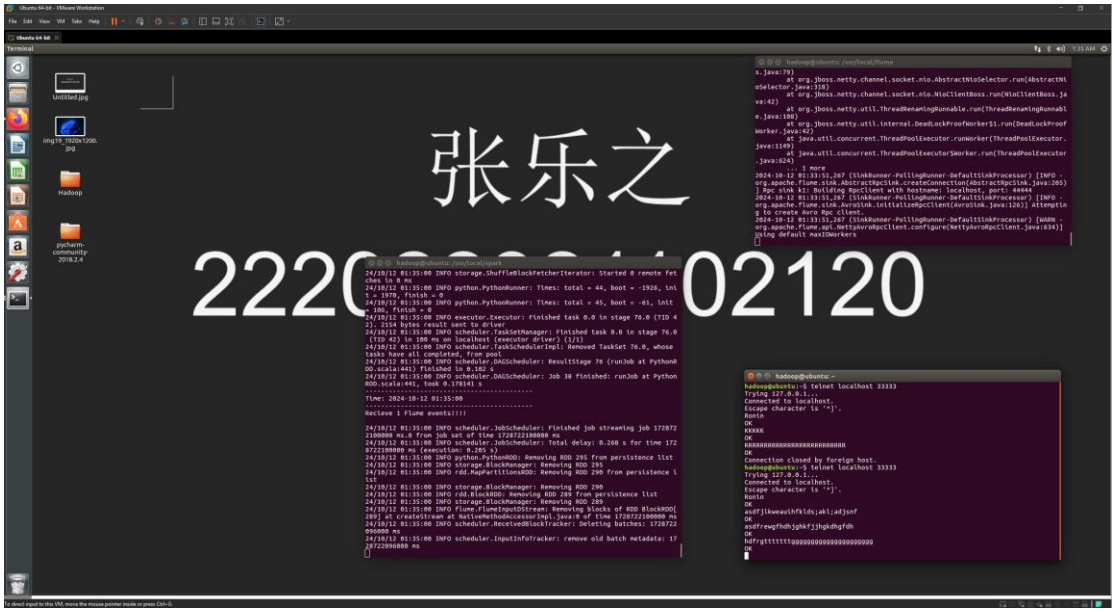
1）下载安装 Flume



2）配置 Flume 数据源

3）编写 Spark 程序使用 Flume 数据源（或直接用 Flume 写入 HDFS 文件，再利用 1 中的文件流监控功能读取 HDFS 文件变化，此方法可以不用重复写 spark streaming 代码）

4）测试监控效果



实验总结：

在本次实验中，我主要进行了基于 Spark Streaming 的流数据处理实践，重点探索了文件流、套接字流以及 Flume 数据源的监控与处理方法。

首先，我编写了一个程序来监控 /home/hadoop/ 目录下文件内容的变化，并实时统计新增文件中的词频。通过 Spark Streaming 的 textFileStream() 方法，我能够监控指定目录下的文本文件，并实时处理其中的文本数据。这种文件流监控机制能够高效地跟踪文件内容的增量变化，进而进行词频统计，充分展示了 Spark Streaming 对文件系统变化的灵活监控能力。

接下来，我实现了套接字流的数据处理任务，编写了一个程序来监控 9543 号端口的实时数据流，并统计传输数据中的词频。通过使用 Spark Streaming 的 socketTextStream() 方法，我能够将端口数据流接入 Spark，并对传输数据进行词频分析。这一实验模拟了实际网络环境中的数据流处理需求，证明了 Spark Streaming 在处理网络流数据方面的强大能力。

实验的最后一部分涉及 Flume 数据源的应用。首先，我成功下载安装并配置了 Flume，将其作为数据流入 Spark Streaming 的数据源。为了实现这一目的，我将数据流入 HDFS，并结合前面编写的文件流监控程序，实时监控 HDFS 文件的变化。这种方式通过 Flume 和 Spark Streaming 的集成，实现了高效的流数据处理，展现了 Flume 在数据收集和传输中的灵活性。

总体而言，这次实验让我深入了解了流数据处理的基本原理与操作方法，特别是 Spark Streaming 在文件流和网络流数据处理中的应用。我还学习了如何通过 Flume 收集数据并与 Spark 集成处理。通过实验，我掌握了处理实时数据的技术与方法，为今后更复杂的流数据处理任务奠定了坚实的基础。