厦门大学林子雨，郑海山，赖永炫 编著

《Spark编程基础（Python版）》

教材配套

机房上机实验指南

实验6 Spark Structured Streaming

（版本号：2020年4月版本）

（答案）



主讲教师：林子雨

厦门大学数据库实验室

二零二零年四月

目录

[一、实验目的 1](#_Toc534215927)

[二、实验平台 1](#_Toc534215928)

[三、实验内容和要求 1](#_Toc534215929)

[1. Syslog介绍 1](#_Toc534215930)

[2. Syslog通过Socket传送到Spark 2](#_Toc534215931)

[3. Syslog日志拆分为DateFrame 2](#_Toc534215932)

[4.对Syslog进行查询 2](#_Toc534215933)

[四、实验报告 7](#_Toc534215934)

[**附录1:任课教师介绍** 7](#_Toc534215935)

[**附录2：课程教材介绍** 8](#_Toc534215936)

[**附录3：高校大数据课程公共服务平台介绍** 9](#_Toc534215937)

厦门大学林子雨,郑海山,赖永炫 编著

《Spark编程基础（Python版）》

第7章 Spark Structured Streaming

教材配套机房上机实验指南

实验6 Spark Structured Streaming编程实践

（答案）

**主讲教师：林子雨**

E-mail: [ziyulin@xmu.edu.cn](mailto:ziyulin@xmu.edu.cn) 个人主页：<http://dblab.xmu.edu.cn/post/linziyu>

# 一、实验目的

（1）通过实验掌握Structured Streaming的基本编程方法；

（2）掌握日志分析的常规操作，包括拆分日志方法和分析场景。

# 二、实验平台

操作系统：Ubuntu16.04

JDK版本：1.8或以上版本

Spark版本：2.4.0

Python版本：3.4.3

数据集：/var/log/syslog

# 三、实验内容和要求

## 1. Syslog介绍

分析日志是一个大数据分析中较为常见的场景。在Unix类操作系统里，Syslog广泛被应用于系统或者应用的日志记录中。Syslog通常被记录在本地文件内，比如Ubuntu内为/var/log/syslog文件名，也可以被发送给远程Syslog服务器。Syslog日志内一般包括产生日志的时间、主机名、程序模块、进程名、进程ID、严重性和日志内容。

## 2. Syslog通过Socket传送到Spark

日志一般会通过kafka等有容错保障的源发送，本实验为了简化，直接将syslog通过Socket源发送。新开一个终端，命令为“tail终端”，输入

tail -n+1 -f /var/log/syslog | nc -lk 9988

tail命令加-n+1代表从第一行开始打印文件内容。-f代表如果文件有增加则持续输出最新的内容。通过管道发送到nc命令起的在本地9988上的服务上。

如果/var/log/syslog内的内容增长速度较慢，可以再新开一个终端，命名为“手动发送log终端”，手动在终端输入

logger ‘I am a test error log message.’

来增加日志信息到/var/log/syslog内。

## 3. Syslog日志拆分为DateFrame

syslog每行的数据类似以下：

Nov 24 13:17:01 spark CRON[18455]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

最前面为时间，接着是主机名，进程名，可选的进程ID，冒号后是日志内容。在Spark内，可以使用正则表达式对syslog进行拆分成结构化字段，以下是示例代码：

# 定义一个偏应用函数，从固定的pattern获取日志内匹配的字段

fields = partial(

regexp\_extract, str="value", pattern="^(\w{3}\s\*\d{1,2} \d{2}:\d{2}:\d{2}) (.\*?) (.\*?)\[\*\d\*\]\*: (.\*)$"

)

words = lines.select(

to\_timestamp(format\_string('2019 %s', fields(idx=1)), 'yy MMM d H:m:s').alias("timestamp"),

fields(idx=2).alias("hostname"),

fields(idx=3).alias("tag"),

fields(idx=4).alias("content"),

)

to\_timestamp(format\_string('2018 %s', fields(idx=1)), 'yy MMM d H:m:s').alias("timestamp"),这句是对Syslog格式的一个修正，因为系统默认的Syslog日期是没有年的字段，所以使用format\_string函数强制把拆分出来的第一个字段前面加上2019年，再根据to\_timestamp格式转换成timestamp字段。在接下来的查询应当以这个timestamp作为事件时间。

## 4.对Syslog进行查询

(1). 统计CRON这个进程每小时生成的日志数，并以时间顺序排列，水印设置为1分钟。

【**参考答案**】

在新开的终端内输入 vi spark\_exercise\_testsyslog1.py ，贴入如下代码并运行。运行之前需要关闭“tail终端”内的tail命令并重新运行tail命令，否则多次运行测试可能导致没有新数据生成。

#!/usr/bin/env python3

from functools import partial

from pyspark.sql import SparkSession

from pyspark.sql.functions import \*

if \_\_name\_\_ == "\_\_main\_\_":

spark = SparkSession \

.builder \

.appName("StructuredSyslog") \

.getOrCreate()

lines = spark \

.readStream \

.format("socket") \

.option("host", "localhost") \

.option("port", 9988) \

.load()

# Nov 24 13:17:01 spark CRON[18455]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

# 定义一个偏应用函数，从固定的pattern获取日志内匹配的字段

fields = partial(

regexp\_extract, str="value", pattern="^(\w{3}\s\*\d{1,2} \d{2}:\d{2}:\d{2}) (.\*?) (.\*?)\[\*\d\*\]\*: (.\*)$"

)

words = lines.select(

to\_timestamp(format\_string('2019 %s', fields(idx=1)), 'yy MMM d H:m:s').alias("timestamp"),

fields(idx=2).alias("hostname"),

fields(idx=3).alias("tag"),

fields(idx=4).alias("content"),

)

# (1). 统计CRON这个进程每小时生成的日志数，并以时间顺序排列，水印设置为1分钟。

windowedCounts1 = words \

.filter("tag = 'CRON'") \

.withWatermark("timestamp", "1 minutes") \

.groupBy(window('timestamp', "1 hour")) \

.count() \

.sort(asc('window'))

# 开始运行查询并在控制台输出

query = windowedCounts1 \

.writeStream \

.outputMode("complete") \

.format("console") \

.option('truncate', 'false')\

.trigger(processingTime="3 seconds") \

.start()

query.awaitTermination()

(2). 统计每小时的每个进程或者服务分别产生的日志总数，水印设置为1分钟。

【**参考答案**】

在新开的终端内输入 vi spark\_exercise\_testsyslog2.py ，贴入如下代码并运行。运行之前需要关闭“tail终端”内的tail命令并重新运行tail命令，否则多次运行测试可能导致没有新数据生成。

#!/usr/bin/env python3

from functools import partial

from pyspark.sql import SparkSession

from pyspark.sql.functions import \*

if \_\_name\_\_ == "\_\_main\_\_":

spark = SparkSession \

.builder \

.appName("StructuredSyslog") \

.getOrCreate()

lines = spark \

.readStream \

.format("socket") \

.option("host", "localhost") \

.option("port", 9988) \

.load()

# Nov 24 13:17:01 spark CRON[18455]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

# 定义一个偏应用函数，从固定的pattern获取日志内匹配的字段

fields = partial(

regexp\_extract, str="value", pattern="^(\w{3}\s\*\d{1,2} \d{2}:\d{2}:\d{2}) (.\*?) (.\*?)\[\*\d\*\]\*: (.\*)$"

)

words = lines.select(

to\_timestamp(format\_string('2019 %s', fields(idx=1)), 'yy MMM d H:m:s').alias("timestamp"),

fields(idx=2).alias("hostname"),

fields(idx=3).alias("tag"),

fields(idx=4).alias("content"),

)

# (2). 统计每小时的每个进程或者服务分别产生的日志总数，水印设置为1分钟。

windowedCounts2 = words \

.withWatermark("timestamp", "1 minutes") \

.groupBy('tag', window('timestamp', "1 hour")) \

.count() \

.sort(asc('window'))

# 开始运行查询并在控制台输出

query = windowedCounts2 \

.writeStream \

.outputMode("complete") \

.format("console") \

.option('truncate', 'false')\

.trigger(processingTime="3 seconds") \

.start()

query.awaitTermination()

(3). 输出所有日志内容带error的日志。

【**参考答案**】

在新开的终端内输入 vi spark\_exercise\_testsyslog3.py ，贴入如下代码并运行。运行之前需要关闭“tail终端”内的tail命令并重新运行tail命令，否则多次运行测试可能导致没有新数据生成。

#!/usr/bin/env python3

from functools import partial

from pyspark.sql import SparkSession

from pyspark.sql.functions import \*

if \_\_name\_\_ == "\_\_main\_\_":

spark = SparkSession \

.builder \

.appName("StructuredSyslog") \

.getOrCreate()

lines = spark \

.readStream \

.format("socket") \

.option("host", "localhost") \

.option("port", 9988) \

.load()

# Nov 24 13:17:01 spark CRON[18455]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)

# 定义一个偏应用函数，从固定的pattern获取日志内匹配的字段

fields = partial(

regexp\_extract, str="value", pattern="^(\w{3}\s\*\d{1,2} \d{2}:\d{2}:\d{2}) (.\*?) (.\*?)\[\*\d\*\]\*: (.\*)$"

)

words = lines.select(

to\_timestamp(format\_string('2019 %s', fields(idx=1)), 'yy MMM d H:m:s').alias("timestamp"),

fields(idx=2).alias("hostname"),

fields(idx=3).alias("tag"),

fields(idx=4).alias("content"),

)

# (3). 输出所有日志内容带error的日志。

windowedCounts3 = words \

.filter("content like '%error%'")

# 开始运行查询并在控制台输出

query = windowedCounts3 \

.writeStream \

.outputMode("update") \

.format("console") \

.option('truncate', 'false')\

.trigger(processingTime="3 seconds") \

.start()

query.awaitTermination()

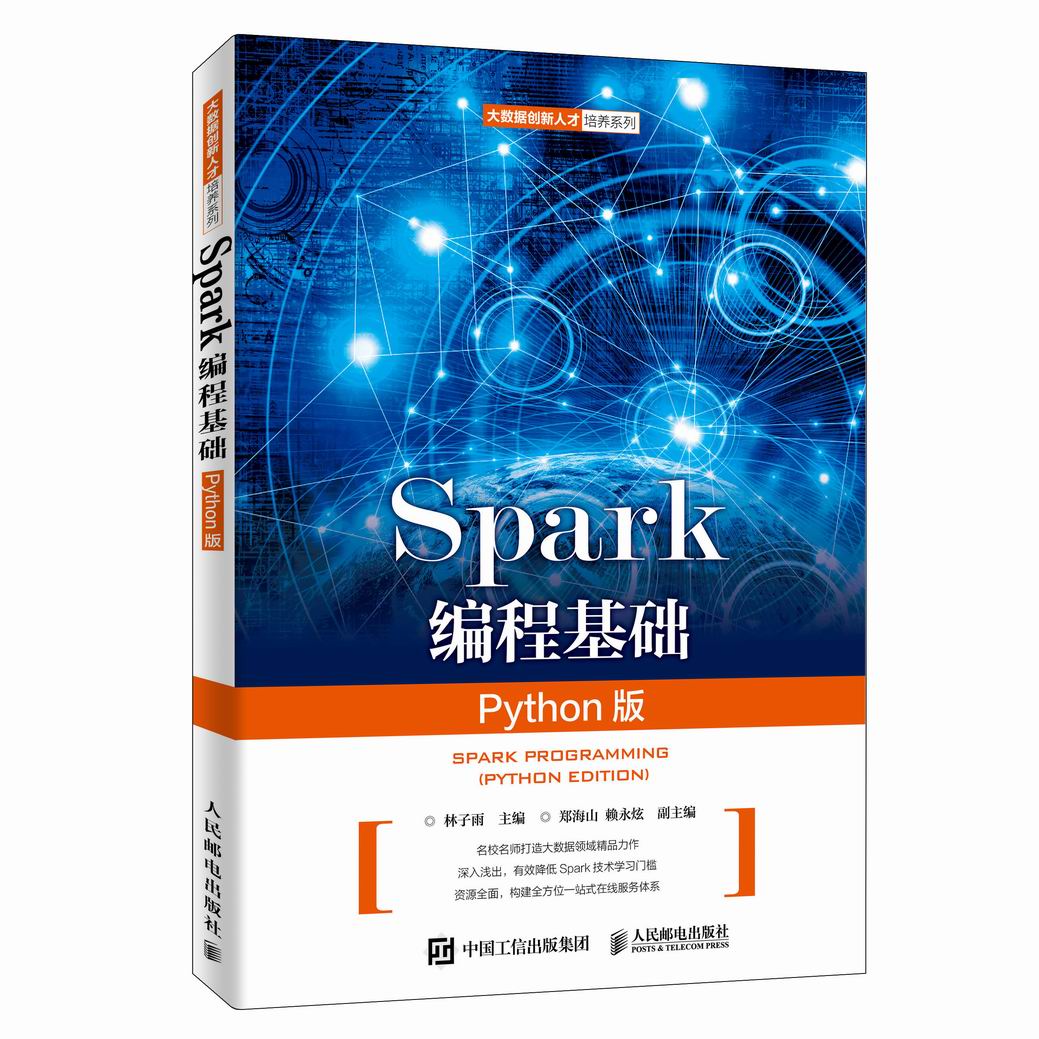
# 四、实验报告

|  |  |  |
| --- | --- | --- |
| 《Spark编程基础（Python版）》实验报告 | | |
| 题目： | 姓名： | 日期： |
| 实验环境： | | |
| 实验内容与完成情况： | | |
| 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）： | | |

**附录1:任课教师介绍**

|  |  |
| --- | --- |
| [http://www.cs.xmu.edu.cn/database/linziyu/images/linziyu2016.jpg](http://dblab.xmu.edu.cn/post/linziyu/) | 林子雨（1978－），男，博士，国内高校知名大数据教师，厦门大学计算机科学系副教授，厦门大学云计算与大数据研究中心创始成员，厦门大学数据库实验室负责人，中国计算机学会数据库专委会委员，中国计算机学会信息系统专委会委员，中国高校首个“数字教师”提出者和建设者。2013年开始在厦门大学开设大数据课程，建设了国内高校首个大数据课程公共服务平台，平台累计网络访问量超过1000万次，成为全国高校大数据教学知名品牌，并荣获“2018年福建省教学成果二等奖”，主持的课程《大数据技术原理与应用》获评“2018年国家精品在线开放课程”。  E-mail: ziyulin@xmu.edu.cn  个人主页：http://dblab.xmu.edu.cn/post/linziyu  数据库实验室网站：http://dblab.xmu.edu.cn |

**附录2：课程教材介绍**



林子雨、郑海山、赖永炫编著《Spark编程基础（Python版）》

人民邮电出版社 ISBN:978-7-115-52439-3 定价：49.80元

厦门大学林子雨、郑海山、赖永炫老师编著《Spark编程基础（Python版）》，以Python作为开发Spark应用程序的编程语言，系统介绍了Spark编程的基础知识。全书共8章，内容包括大数据技术概述、Spark的设计与运行原理、Spark环境搭建和使用方法、RDD编程、Spark SQL、Spark Streaming、Structured Streaming、Spark MLlib等。本书每个章节都安排了入门级的编程实践操作，以便读者更好地学习和掌握Spark编程方法。本书官网免费提供了全套的在线教学资源，包括讲义PPT、习题、源代码、软件、数据集、授课视频、上机实验指南等。

本书可以作为高等院校计算机、软件工程、数据科学与大数据技术等专业的进阶级大数据课程教材，用于指导Spark编程实践，也可供相关技术人员参考。

欢迎访问《Spark编程基础(Python版)》教材官方网站：http://dblab.xmu.edu.cn/post/spark-python/



扫一扫访问教材官网

**附录3：高校大数据课程公共服务平台介绍**



高校大数据课程公共服务平台，由中国高校首个“数字教师”的提出者和建设者——林子雨老师发起，由厦门大学数据库实验室全力打造，由厦门大学云计算与大数据研究中心携手共建。这是国内第一个服务于高校大数据课程建设的公共服务平台，旨在促进国内高校大数据课程体系建设，提高大数据课程教学水平，降低大数据课程学习门槛，提升学生课程学习效果。平台服务对象涵盖高校、教师和学生。平台为高校开设大数据课程提供全流程辅助，为教师开展教学工作提供一站式服务，为学生学习大数据课程提供全方位辅导。平台重点打造“11个1工程”，即1本教材（含官网）、1个教师服务站、1个学生服务站、1个公益项目、1堂巡讲公开课、1个示范班级、1门在线课程、1个交流群（QQ群、微信群）、1个保障团队、1个培训基地和1个实验平台。目前平台每年访问量已经超过200万次，累计访问量超过1000万次，成为国内高校大数据教学知名品牌。

平台主页：http://dblab.xmu.edu.cn/post/bigdata-teaching-platform/



扫一扫访问平台主页