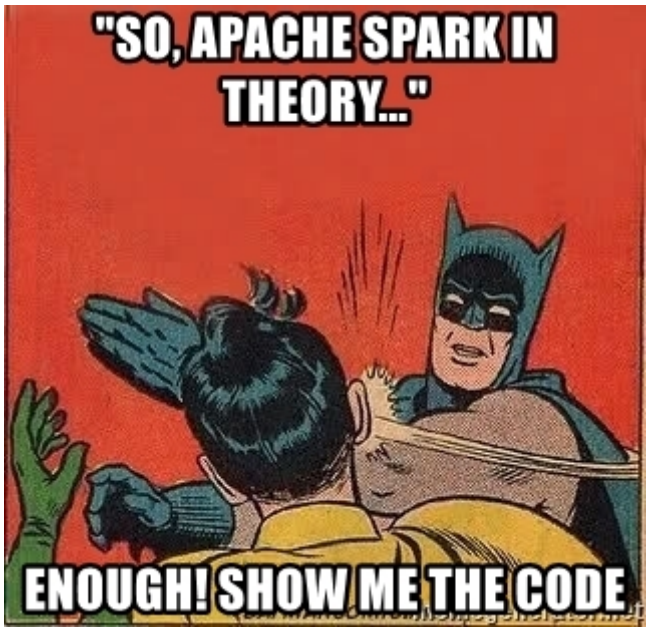
实验三 Spark



零、服务器集群&Spark注意事项

1.可使用的服务器集群

从lab2开始，我们上线了新的服务器集群，目前有两个集群(两个集群不互通，大家可以选择随意选择在集群一或二上实验^ ^)：

集群一：ip地址：`10.103.9.11`，可用机器：01，03-06。登录集群一01的命令：`ssh xxx@10.103.9.11` （也就是和以下实验指导书的内容完全相同）

集群二：ip地址：`10.103.10.156`，可用机器：01-04。登录集群二01的命令：`ssh xxx@10.103.10.156 -p 8001`（**由于进入该集群的端口并非默认端口，所以在 `ssh` 指令后面一定要用 `-p` 要加上端口号！**）

其中 xxx为学号，默认密码也为学号。

推荐学号尾号为奇数的同学使用集群一，学号尾号为偶数的同学使用集群二，以减轻服务器集群的负担。

2.实验中的注意事项

很多同学在使用完`spark-shell`后直接关闭窗口或`ctrl+z`挂起，未正常退出，导致spark进程仍在运行，使得服务器资源不足。这就会导致其他同学在进行实验时出现以下情况：

```
textFile: org.apache.spark.rdd.RDD[String] = /dsjxtjc/rwlu/wc_dataset.txt MapPartitionsRDD[1] at textFile at <console>:24

scala> textFile.first()
[Stage 0:>                                                          (0 + 0) / 1]23/08/13 22:49:04 WARN TaskSchedulerImpl: Initial job has not accept
ed any resources; check your cluster UI to ensure that workers are registered and have sufficient resources
23/08/13 22:49:19 WARN TaskSchedulerImpl: Initial job has not accepted any resources; check your cluster UI to ensure that workers are registered an
d have sufficient resources
23/08/13 22:49:34 WARN TaskSchedulerImpl: Initial job has not accepted any resources; check your cluster UI to ensure that workers are registered an
d have sufficient resources
```

所以大家在正常完成实验之后，请务必使用 `quit()` 正常退出 `spark` 进程。

一、实验目标

本次实验旨在使用Spark 处理数据并实现机器学习算法。具体如下：

- 学习使用Spark-shell 基本命令（1 分）；
- 使用Spark 实现词频统计（4 分）；
- 使用Spark 实现线性回归训练算法（5 分）。

参考资料

- RDD Programming Guide https://spark.apache.org/docs/latest/rdd-programming-guide.html
- Spark https://www.usenix.org/system/files/conference/nsdi12/nsdi12-final138.pdf

二、学习Spark-shell常用指令

基本功能介绍

Spark-shell 是一个强大的交互式分析数据工具，同时也是一种学习Spark的有效工具，它可以使用Scala或Python编写，本课程主要介绍Scala。开启spark-shell：

```
2020214210@thumm01:~$ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark Context Web UI is available at Spark Master Public URL
Spark context available as 'sc' (master = spark://thumm01:7077, app id = app-20211017105714-0002).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.4
      /_/

Using Scala version 2.12.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_221)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

Spark-shell 除了可以使用Scala 语言操作外，还有一些基本指令，这些指令都以 `:` 开头，指令的用法可以使用 `:help` 查看：

```
scala> :help
All commands can be abbreviated, e.g., :he instead of :help.
:completions <string>    output completions for the given string
:edit <id>|<line>        edit history
:help [command]          print this summary or command-specific help
:history [num]           show the history (optional num is commands to show)
:h? <string>             search the history
:imports [name name ...] show import history, identifying sources of names
:implicits [-v]          show the implicits in scope
:javap <path|class>      disassemble a file or class name
:line <id>|<line>        place line(s) at the end of history
:load <path>             interpret lines in a file
:paste [-raw] [path]     enter paste mode or paste a file
:power                   enable power user mode
:quit                    exit the interpreter
```

```
:replay [options]         reset the repl and replay all previous commands
:require <path>           add a jar to the classpath
:reset [options]          reset the repl to its initial state, forgetting all
session entries
:save <path>              save replayable session to a file
:sh <command line>        run a shell command (result is implicitly =>
List[String])
:settings <options>       update compiler options, if possible; see reset
:silent                   disable/enable automatic printing of results
:type [-v] <expr>         display the type of an expression without
evaluating it
:kind [-v] <type>         display the kind of a type. see also :help kind
:warnings                 show the suppressed warnings from the most recent
line which had any
```

其中，常用指令为：

- `:quit` 退出spark-shell控制台；
- `:load <path>` 加载使用scala编写的spark-shell脚本；
- `:save <path>` 将当前上下文的历史指令保存为文件。

使用`:load`打印Hello Word（1 分）

在ubuntu自己的目录下新建一个Scala文件，写入`println("HelloWorld!")`，在此目录下进入`spark-shell`使用`:load <path>`运行该文件。（本题 1 分）

三、使用Spark 进行词频统计

0.词频统计要求

以下要求对下述两个任务都适用。

输出格式

为了方便进行正确性的验证，请将你词频统计的输出结果保证按照ASCII字典序从小到大的格式进行输出。每行一个按照 `<word> <count>` 的形式输出，代表该词汇出现的次数。

以 `wc_dataset.txt` 为例，你需要保证输出结果为：

```
'tis 5
Again 5
Alabama 15
Alleghenies 5
Almighty 5
America 25
American 20
And 60
But 20
California 5
Carolina 5
Catholics 5
Colorado 5
Constitution 5
Continue 5
Declaration 5
Emancipation 5
```

```
Five 5
Free 5
From 10
Gentiles 5
Georgia 15
Go 5
God 5
God's 15
Hampshire 5
Happiness 5
I 75
In 10
Independence 5
Instead 5
It 20
Jews 5
Land 5
Let 45
Liberty 5
Life 5
Lookout 5
Lord 5
Louisiana 5
Mississippi 20
Mountain 10
My 5
Negro 65
Negro's 5
New 15
Nineteen 5
No 5
Now 25
One 15
Pennsylvania 5
Pilgrim's 5
Proclamation 5
Protestants 5
Rights 5
Rockies 5
Some 5
South 10
Stone 5
Tennessee 5
Thank 5
The 10
There 5
This 25
We 50
When 10
With 15
York 10
You 5
a 63875
abide 100
able 140
... // 后续词汇较多，予以省略
```

提交报告时附带中间过程文件

你需要在你提交的实验报告压缩包当中提供你运行词频统计期间的文件夹，包括以下内容（其中文件夹命名为原始实验记录，不需要与此处相同）

```
output
|-- _SUCCESS
|-- part-00000
|-- part-00001
|-- ...
```

1. map+reduceByKey（2分）

本题请同学们使用自己的数据集，这里需要**先将数据集传入Hadoop文件系统**（见实验二）。请将输出结果放入实验报告

首先，进入spark-shell

```
2020214210@thumm01 :~$ spark-shell
```

接下来，我们需要加载待统计词频的数据集，输入以下内容：

```
scala> val words = sc.textFile("/dsjxtjc/2020214210/wc_dataset.txt")
words: org.apache.spark.rdd.RDD[String] = /dsjxtjc/2020214210/wc_dataset.txt
MapPartitionsRDD[1] at textFile at <console>:24
```

这句命令中，`sc`（Spark-Context）是spark-shell的上下文，这个变量是进入spark-shell就有的，可以用来设置一些运行参数；`val words` 是定义一个变量名为words的变量，它的值是使用`sc.textFile`函数加载HDFS 中words.txt文件的内容。接下来查看words的内容：

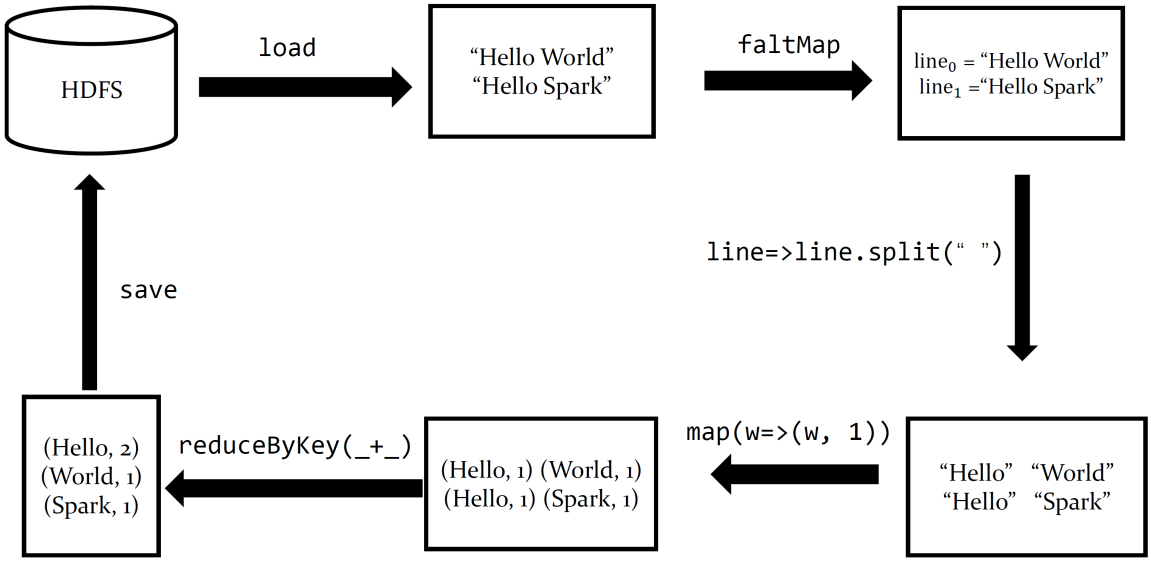
```
scala > words.first() # 查看第一行
res0: String = chapter
scala > words.count() # 查看行数
res1: Long = 2683500
```

使用一行代码统计词频：

```
scala > val result = words.flatMap(l => l.split(" ")).map(w => (w,
1)).reduceByKey(_ + _)
result: org.apache.spark.rdd.RDD[(String , Int)] = ShuffledRDD [4] at
reduceByKey at <console>:25
scala > result.first() # 查看结果的第一行内容
res3: (String, Int) = (someone, 100)
scala > result.saveAsTextFile("/dsjxtjc/2020214210/wc_output")
```

接下来我们来解释这行代码，代码的流程如下图所示，可分为以下步骤：

- 对`words`逐行处理，对每一行按空格进行分割，得到一个字符串列表；
- 使用`map`将字符串列表转成一个键值对列表`[<key1, value1>, <key2, value2>, ……]`，其中键为单词，值为词频（没有合并之前为1）；
- 将不同的键值对根据相同的键不断地合并，直至无法合并，得到词频统计结果；
- 将结果保存到HDFS 中(保存到了`/dsjxtjc/student_id/wc_output`)。



2. 换用另一种RDD函数组合实现WordCount功能，测试不同实现方式的所需时间。（2分）

- 时间：`var t = new Date().getTime`

四、Spark 机器学习

请生成自己的数据集，利用Spark实现线性回归。禁止使用任何MLlib库函数。（本题 5 分）

1. 一元线性回归给 2 分；
2. 多元线性回归（非向量运算，即使用循环更新参数）给 3 分；
3. 向量表示的多元线性回归给 5 分。

提示：

- 线性回归 https://zhuanlan.zhihu.com/p/72513104
- 循环实现和向量实现 https://zhuanlan.zhihu.com/p/154015989
- 以上三点为三种实现方式，实现任意一种即可
- 可以包括矩阵加法、矩阵减法、矩阵乘法、矩阵转置等接口；
- 需要截图放出每一轮的loss，如果训练轮次太多，不一定要全部截图，只需要展示它在下降即可；

五、报告提交要求

请严格按照以下要求提交实验报告。

1. 将命令、关键代码（文本）、结果截图放入报告，实验报告需为pdf 格式，连同代码文件一同打包成压缩文件（命名为`学号_姓名_实验一.*`，例如：`2021200000_张三_实验一.zip`），最后提交到网络学堂。压缩文件中文件目录应为：

```
.
└── 学号_姓名_实验二.pdf
└── code
    └── code_file1.py
    └── code_file2.py
    └── ...
```

2. 迟交作业一周以内，以50% 比例计分；一周以上不再计分。另一经发现抄袭情况（包括往届），零分处理。