

（深圳）

实验报告

开课学期： 2022春季

课程名称： 大数据计算基础

实验名称： 健康码红码生成模拟

实验性质： 设计型

实验学时： 2 地点： T2210

学生学号： 200810216

学生姓名： 李墨

实验与创新实践教育中心制

2022年1月

1. **实验环境**

Spark编程方案：

（1） 使用CentOS 6.10的Linux操作系统。

（2） 使用1.8版本的JDK。

（3） 使用2.6.4版本的Hadoop。

（4） 使用2.3.1版本的Spark。

（5） 使用IntelliJ IDEA作为编程软件。

Python编程方案：

（1） 使用Windows11的操作系统。

（2） 使用3.7.13版本的Python。

（3） 使用Google Colaboratory及Python作为编程工具。

（4） 使用关键Python Packages如下：

dask 版本2.12.0

pandas 版本1.3.5

1. **小组方案及结果简述**

*请描述实现方案的主要内容以及步骤，对实验结果做简单描述（不需要贴文件，只需要对结果做些述，比如找出多少个需要标记红码的手机号，耗时多久，做没做提升阶段等内容）*

实验方案：

方案一：

1. 采用大数据平台Spark，通过IntelliJ作为编译器，Scala作为编程语言，提交到由三个结点组成的Yarn集群上运行。
2. 方案步骤：
3. 终端测试scala命令
4. 查阅大量Spark编程资料
5. IntelliJ上编写代码
6. 打包成jar包提交到集群运行

最终实验当天步骤：

1. 下载数据集，上传Spark源代码
2. 上传文件到HDFS
3. 在终端提交Spark-submit命令

（3）方案结果与耗时：

在同时做基础任务和提升任务的情况下：

测试数据（20MB的txt文件）耗时：1min

正式数据（8GB的txt文件）实际净耗时：20min左右，找出了91964行的redmark，10705行的superredmark。

（4）大致思路：读入txt文件后，转化为Spark的dataframe，通过判断每一行数据是否处于被污染基站，运用Spark的filter语句，去重排序后得出redmark.txt。在redmark基础上，运用两个列表分别记录被污染基站的污染起始时间与污染终止时间。通过使用用户自定义函数UDF进一步过滤dataframe得到superredmark。

方案二：

（1）采用基于大型数据集并行处理框架Dask的Python处理方法，并发处理分块的数据集， 在本地笔记本电脑多线程并发运行。

（2）方案耗时

在同时做基础任务和提升任务的情况下：

测试数据耗时：小于1秒

正式数据耗时：40秒左右，找出了91964行的redmark，10705行的superredmark

（3）大致思路：大体上与Spark思路相同，只是采用了不同的大数据处理架构Dask，分块读取cdinfo.txt文件并行处理。此外在判断superredmark时去除了Spark中的全列表扫描，而是运用Python中极为灵活的字典类型来大幅度缩短搜索时间。

**比赛正式开始时，先提交了Spark任务的命令，在Spark运行时同时进行数据集到本地的下载。两种方法同时统筹安排进行，通过对比Python与Spark的结果，验证无误后提交结果。**

1. **个人参与部分**

通过查询大量资料，完成了Spark平台Scala脚本的编写，其中包含基础任务和提升任务，同时在最后一天，觉察到提供的大数据平台单个节点性能欠佳之后，**且平台存在不稳定性。**考虑到比赛的**竞速性质**，决定用Python作为备用解决本次问题，在尝试过pandas读取大文件数据集会造成程序**直接崩溃**之后，查询广泛资料，得到python中大数据处理的包——Dask Dataframe的相关资料：

*“Dask DataFrame 是一个大型并行 DataFrame，由许多较小的 Pandas DataFrames 组成，沿索引拆分。这些 Pandas DataFrames 可能存在于磁盘上，用于在单个机器上或在集群中的许多不同机器上进行大于内存的计算。一个 Dask DataFrame 操作会触发对组成 Pandas DataFrames 的许多操作。”*

并通过查询Dask Dataframe的常见API，不仅实现了基础任务，还实现了进阶任务，并相比在Spark平台上处理大型数据集的时间缩短了至少30倍。

正式比赛开始前，编写Python脚本，来生成大数据集（考虑到老师只发了一个20MB的，担心正式实验开始时，由于数据量太大，某些部分会产生bug），最终生成了一个8GB的预测文件，将其分别用Spark及Dask处理，均处理无误。正常运行。

正式比赛开始时，由于平台的不稳定性，导致本小组延迟21分钟进入实验平台。我负责了对正式数据集的下载，负责了包含redmark及superredmark在内的结果的生成。在做完基础任务之后，代码运行出现了问题，在组员的协同帮助下，有条不紊的完成了代码的修改，经过分析，这是由于测试数据集中的数据没有出现过同一个人进进出出同一个基站的特殊情况，导致我的superredmark代码无法正常运行，经过20分钟的仔细修改，最终修改完毕，增强了代码的鲁棒性，仍带领小组保持第一的名次上传了superredmark的结果。

上传完进阶任务的结果后，全部组员都开始了对最终结果的核查，以检查最终结果会不会出现问题，在各个组员的认真联合核查下，未发现任何错误。经过对照别的组后来发的基础任务的结果，和我们组的基本完全一致，于是其余组员将结果检查工作最终将重心放在进阶任务superredmark的核查上。截至到最后，未发现结果数据中的任何问题。

1. **心得体会**

关于本次实验，个人认为之所以基于内存的大数据平台Spark处理的速度不如本地Python快的原因如下，

首先，最终使用的Python脚本并非普通的代码，通常我们用Python处理文件会使用Pandas包来转化为dataframe处理数据，但如果在本次实验中直接用Pandas dataframe，即使是简单的dataframe.loc语句都无法正常执行，经过查阅得知Pandas本身就是基于内存的数据处理，当数据量太大时候，即使是更新后的测试数据50万条数据，也无法正常运行（因为Pandas需要创建一个50万行的numpy数组，这会直接超出一般笔记本的大小，**更不用说**最终正式数据中的2亿条数据），查阅大量资料后，查询到面向大数据集的处理方式——Dask的存在，并成功在最终8个GB的数据集分为80个块，每个block大小为100MB左右，利用多线程并行，快速的完成了任务。

其次， 是大数据平台分发的资源有限，毕竟要分给很多同学使用，即使是共有三个节点，但每个节点的性能欠佳，每个节点据老师说只有1G的内存（实际我在Spark控制的网页上看每个结点只有512MB，可能是环境不太一样），即使三个结点加起来也不如一台普通的笔记本电脑的内存大（实际最终运行的笔记本电脑的内存是16GB）。

**（4）关于Dask 与Spark的比较**，[官网上此处](https://docs.dask.org/en/stable/spark.html?highlight=spark#comparison-to-spark)可以查得详细资料，关键差别列举在下面，方便审阅者和我自己更加了解两者的差异：

总结：通常 Dask 比Spark更小更轻。这意味着它的功能更少，而是与其他库一起使用，尤其是数字 Python 生态系统中的库。它与 Pandas 或 Scikit-Learn 等库结合使用以实现高级功能。

语言：Spark是用 Scala 编写的，对 Python 和 R 有一些支持。它与其他 JVM 代码的互操作性很好。Dask 是用 Python 编写的，并且**只真正支持 Python**。它可以与 C/C++/Fortran/LLVM 或其他通过 Python 链接的本机编译代码很好地互操作。

生态系统：Spark是一个多合一的项目，它激发了自己的生态系统。它与许多其他 Apache 项目很好地集成在一起。Dask 是更大的 Python 生态系统的一个组成部分。它结合并增强了 NumPy、Pandas 和 Scikit-Learn 等其他库

年龄和信任；Spark较旧（自 2010 年以来），已成为大数据企业世界中占主导地位且备受信赖的工具。Dask 更年轻（自 2014 年以来），是备受信赖的 NumPy/Pandas/Scikit-learn/Jupyter 堆栈的扩展。

范围：Spark更专注于传统的商业智能操作，如 SQL 和轻量级机器学习。Dask 更广泛地应用于商业智能应用程序以及许多科学和**自定义情况**。

**数组**：Spark本身**不包括对多维数组的支持**，尽管在 MLLib 中可以找到对二维矩阵的一些支持。人们可能还想看看 Thunder项目，它结合了 Apache Spark 和 NumPy 数组。然而Dask 完全支持可扩展多维数组的 NumPy 模型。

（5）**个人体会**：Scala语言写起来，相比于Python容错性较差，很多时候想要执行一些简单的操作，对于Scala新手写起来都是极其麻烦的（仅代表个人意见）。例如Scala的原始List甚至无法在原来的列表中添加元素，只能在原来基础上创一个新的List，以及RDD算子无法在for循环内运行（我与组员谭启泰debug好久才得知此问题，原因是RDD算子惰性，如果在for循环内部积累太多不执行，会最终导致栈溢出）。Scala使用起来对于不熟悉Scala语言的人来说各种要求苛刻了。初学者写起来每一步都是bug。个人写Scala语言实现基本任务和进阶任务共写了3天，Python的Dask语言只通宵了一夜加上一上午debug就完成了。虽然Spark固有其自身的优势，其与目前的各种Apache项目结合更加紧密。在实际上的大公司中可能更适合使用Spark，而Dask实现起来用户自定义功能更加方便。

（6）模型展望：

实际上，本次大作业数据量极大、操作比较单一，感觉很适合用GPU来进行，我曾考虑过使用GPU加速，毕竟GPU加速有时常常可以提升100倍的速度，我查阅了大量资料，得知cuda推出过一个cudf(即cuda dataframe)可以使用CUDA对大数据集进行GPU加速处理，此外numba也是一个可以用来个别python代码的Python Package。**也许**可以将本任务执行时间进一步从我们组的用时40秒缩小到几秒钟？由于以下原因暂时没有做出来，这可能是本次实验可以进一步走向完美的地方，：

一、cudf的安装要求极其高，不仅要求CUDA、Python版本，还要求各种使用的依赖包的版本，常常需要使用conda来安装。按照官方给的操作步骤没有安装成功。而numba虽然可以安装，但是其对很多python代码**不支持**GPU加速。

二、时间限制：在正式数据发布前我们刚刚debug完Python利用Dask dataframe做的。并且时间已经相比Spark平台加速了几乎30倍。相比别的组来说时间上已经取得巨大优势。暂时没有进一步提升性能的必要。

1. **附上代码（文本式）**

%Spark平台的Scala代码如下

**%虽然最终的Scala总行数仅仅不到100行，然而凝结在Scala每一行代码背后的资料查阅过程是更加宝贵的。常常每一行Scala代码背后都要查几个网页来学习。学习完之后几乎每一行代码都要debug**

package Experiment4

import org.apache.spark.sql.functions.{col, \_}

import org.apache.spark.sql.{Row, SparkSession}

object Experiment4 {

def main(args: Array[String]): Unit = {

//建立Spark连接

val spark = SparkSession.builder().appName("Experiment4").getOrCreate()

val sc = spark.sparkContext

// 将txt文件按照csv格式读入

val cdinfo = spark.read.csv(args(0))

val infected = spark.read.csv(args(1))

val infected\_tlp=infected.select("\_c0").rdd.map(row => row(0)).collect.toList.distinct

// 找出感染的基站

val filtered = cdinfo.filter(cdinfo("\_c3").isin(infected\_tlp:\_\*))

val infected\_base\_num\_ori=filtered.select("\_c0").rdd.map(row => row(0)).collect.toList

// 基站去重

var infected\_base\_num = List[Any]()

for (i <- 0 to (infected\_base\_num\_ori.length-1)){

if(i%2==1){

infected\_base\_num = infected\_base\_num :+ infected\_base\_num\_ori(i)

}

}

// 使用两个列表记录基站开始及结束的污染时间

val infected\_base\_start = filtered.filter(col("\_c2")===1).select("\_c1").rdd.map(row => row(0)).collect.toList

val infected\_base\_end = filtered.filter(col("\_c2")===2).select("\_c1").rdd.map(row => row(0)).collect.toList

val infected\_base=infected\_base\_num.distinct

val infected\_people=cdinfo.filter(cdinfo("\_c0").isin(infected\_base:\_\*))

val may\_infected = infected\_people.groupBy("\_c0","\_c3").agg(collect\_list("\_c1").as("time"))

may\_infected.show(false)

// 使用UDF（用户自定义函数）过滤

def is\_infected( r:Row ) : Boolean = {

for (i <- 0 to (infected\_base\_num.length-1))

{

if (r.getAs("\_c0")==infected\_base\_num(i))

{

if (((r.getAs[Seq[String]]("time"))(0).toLong <= infected\_base\_end(i).toString.toLong ) && ((r.getAs[Seq[String]]("time"))(1).toLong >= infected\_base\_start(i).toString.toLong))

{

return true;

}

}

}

return false;

}

// 注册用户自定义函数

spark.udf.register("is\_infected",is\_infected \_)

val final\_infected = may\_infected.filter(callUDF("is\_infected",struct(may\_infected.columns.map(may\_infected(\_)) : \_\*)))

final\_infected.show()

// 将结果写入txt文件

val basic\_task=infected\_people.select(("\_c3")).distinct().sort("\_c3")

basic\_task.coalesce(1).write.text(args(2))

val final\_task= final\_infected.select(("\_c3")).distinct().sort("\_c3")

final\_task.coalesce(1).write.text(args(3))

}

}

#Python调用Dask API实现的代码如下

# -\*- coding: utf-8 -\*-

"""Dask\_final.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1ucWbtHajnzDkX3s\_UfJ92Mm1iAr-HWFs

"""

# !pip install dask

# !pip install dask[dataframe] --upgrade

#导入相关的包

import pandas as pd

import dask.dataframe as dd

from timeit import default\_timer as timer

# 记录代码运行时间

start = timer()

# 使用大数据处理框架Dask分块读取txt文件

cdinfo = dd.read\_csv('cdinfo.txt', blocksize=100e6, names=['base', 'time', 'inorout', 'telephone'])

# cdinfo = pd.read\_csv('cdinfo.txt',names = ['base','time','inorout','telephone'])

# pandas不能正常读取大数据集文件，即使成功读取，也极容易在操作中内存不足而崩溃

infected = pd.read\_csv('infected/infecteda03.txt', names=['telephone'])

infected\_list = list(set(infected["telephone"].tolist()))

cdinfo\_infected = cdinfo.loc[cdinfo["telephone"].isin(infected\_list)]

# 感染列表转为pandas dataframe(考虑到只有10条数据)

cdinfo\_infected = cdinfo\_infected.compute()

infected\_base = list(set(cdinfo\_infected["base"]))

infected\_base.sort()

# 如果某人在感染基站待过，标记为redmark

redmark\_df = cdinfo.loc[cdinfo["base"].isin(infected\_base)].compute()

redmark = list(set(redmark\_df["telephone"].tolist()))

redmark.sort()

# 写入redmark.txt

with open('redmark/redmarkA03.txt', 'w') as fp:

for item in redmark:

# write each item on a new line

fp.write("%s\n" % item)

print('Done')

# superredmark部分

infected\_base\_time = cdinfo\_infected.groupby(["base", "telephone"])["time"].apply(list).reset\_index( name='infected\_time').drop("telephone", 1)

# 对于每一个被污染的基站，通过字典记录其被污染的时间段

# 通过使用字典而不使用列表而避免暴力搜索，大幅度加快寻找时间

infected\_base\_time\_dict = {}

for index, row in infected\_base\_time.iterrows():

if row["base"] not in infected\_base\_time\_dict.keys():

infected\_base\_time\_dict[row["base"]] = [row["infected\_time"]]

else:

infected\_base\_time\_dict[row["base"]].append(row["infected\_time"])

redmark\_df = redmark\_df.groupby(["base", "telephone"])["time"].apply(list).reset\_index(name='stay\_time')

# 以下代码在正式数据中临时加入

# 以避免某人去了某一基站两次

stay\_time = redmark\_df['stay\_time'].tolist()

fmax = 0

for i in range(len(stay\_time)):

stay\_time[i].sort()

if len(stay\_time[i]) >= fmax:

fmax = len(stay\_time[i])

print("fmax", fmax)

# fmax=4,意味着存在某人进出进出某相同基站

# 将dataframe拆分

split\_redmark\_df = pd.DataFrame(stay\_time, columns=['start', 'end', "start1", "end1"])

# split\_redmark\_df = pd.DataFrame(redmark\_df['stay\_time'].tolist(),columns=['start','end'])

# 连接两个dataframe

redmark\_df = pd.concat([redmark\_df, split\_redmark\_df], axis=1).drop("stay\_time", 1)

is\_infected = []

# 判断被标记为redmark的人是否能够被标记为superredmark

# 通过从redmark中继续筛选，而不是从cdinfo.txt中重新查找，代码运行时间将被大幅缩短

for index, row in redmark\_df.iterrows():

flag = False

for i in infected\_base\_time\_dict[row["base"]]:

if i[1] >= row["start"] and i[0] <= row["end"]:

flag = True

if i[1] >= row["start1"] and i[0] <= row["end1"]:

flag = True

is\_infected.append(flag)

superredmark\_df = redmark\_df.loc[is\_infected]

final = list(set(superredmark\_df["telephone"].tolist()))

final.sort()

# 写入superredmark.txt

with open('superredmark/superredmarkA03.txt', 'w') as fp:

for item in final:

# write each item on a new line

fp.write("%s\n" % item)

print('Done')

print("共运行时间" + f"{timer() - start}")

1. **部分参考资料**

[Dask](https://docs.dask.org/en/stable/)

[Dask – How to handle large dataframes in python using parallel computing](https://www.machinelearningplus.com/python/dask-tutorial/)

[pass whole Row to UDF - Spark DataFrame filter](https://stackoverflow.com/questions/31816975/how-to-pass-whole-row-to-udf-spark-dataframe-filter)

[Scala Basic Tutorial - How To Declare Variables And Types](http://allaboutscala.com/tutorials/chapter-2-learning-basics-scala-programming/scala-basic-tutorial-declare-variables-types/)

[scala.collection.mutable.WrappedArray$ofRef cannot be cast to Integer](https://stackoverflow.com/questions/40199507/scala-collection-mutable-wrappedarrayofref-cannot-be-cast-to-integer)

[String object to Integer type](https://stackoverflow.com/questions/36547375/how-to-resolve-java-lang-string-cannot-be-cast-to-java-lang-integer-android-with)

[print type of variable in Scala](https://stackoverflow.com/questions/36700010/print-type-of-variable-in-scala):Object.getClass

[Java String to long Example](https://www.javatpoint.com/java-string-to-long#:~:text=%E2%86%92%20%E2%86%90%20prev-,Java%20String%20to%20long,parseLong()%20method.) However,this is for Java,not for scala

[not serializable exception](https://stackoverflow.com/questions/22592811/task-not-serializable-java-io-notserializableexception-when-calling-function-ou):so I change anonymous function to others

[Cannot convert string to a long in scala](https://stackoverflow.com/questions/41215421/cannot-convert-string-to-a-long-in-scala):not works for me ,but helpful

[overloaded method value filter with alternatives](https://stackoverflow.com/questions/37334915/dataframe-error-overloaded-method-value-filter-with-alternatives)

[Spark UDFs, how to write them and some gotchas?](https://medium.com/@achilleus/spark-udfs-we-can-use-them-but-should-we-use-them-2c5a561fde6d)

[How to pass whole Row to UDF - Spark DataFrame filter](https://stackoverflow.com/questions/31816975/how-to-pass-whole-row-to-udf-spark-dataframe-filter)

[学校平台课程](http://bdes.hitsz.edu.cn:50080/course/view.php?id=120&section=1)

[scala for loop](https://www.tutorialspoint.com/scala/scala\_for\_loop.htm)

[How to Sort DataFrame column explained](https://sparkbyexamples.com/spark/spark-how-to-sort-dataframe-column-explained/)

[Spark DataFrame Where Filter | Multiple Conditions](https://sparkbyexamples.com/spark/spark-dataframe-where-filter/)

[Provide schema while reading csv file as a dataframe](https://stackoverflow.com/questions/39926411/provide-schema-while-reading-csv-file-as-a-dataframe):inferSchema

[Spark Read CSV file into DataFrame](https://sparkbyexamples.com/spark/spark-read-csv-file-into-dataframe/)

[value $ is not a member of StringContext](https://stackoverflow.com/questions/44209756/value-is-not-a-member-of-stringcontext-missing-scala-plugin)

[How to print a list in Scala](https://www.geeksforgeeks.org/how-to-print-a-list-in-scala/)

[Convert List[Any] to List[Int]](https://stackoverflow.com/questions/22442470/convert-listany-to-listint)

[collect\_list](https://spark.apache.org/docs/latest/api/sql/index.html)

[display full value of column in spark-shell](https://stackoverflow.com/questions/37741841/fetch-more-than-20-rows-and-display-full-value-of-column-in-spark-shell)

[Function Definitions](https://www.tutorialspoint.com/scala/scala\_functions.htm)

[Spark TaskNotSerializable when using anonymous function](https://stackoverflow.com/questions/32892995/spark-tasknotserializable-when-using-anonymous-function)

[pyspark.sql.functions.flatten](https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.sql.functions.flatten.html)

[Spark SQL UDF (User Defined Functions)](https://sparkbyexamples.com/spark/spark-sql-udf/)

[Spark-SQL-functions](https://sparkbyexamples.com/spark/spark-sql-functions/)

[Filter with anonymous functions](https://stackoverflow.com/questions/31816975/how-to-pass-whole-row-to-udf-spark-dataframe-filter)

[Get item in the list in Scala?](https://stackoverflow.com/questions/4981689/get-item-in-the-list-in-scala)

[Spark Groupby Example with DataFrame](https://sparkbyexamples.com/spark/using-groupby-on-dataframe/)

[Merge multiple spark rows to one](https://stackoverflow.com/questions/60003311/merge-multiple-spark-rows-to-one)

[Scala Tutorial 24 - map, flatMap, flatten and filter (Higher-order Methods)](https://www.youtube.com/watch?v=K5alCbt6MoQ)

[Map vs flat Map| Spark Transformation | Spark Tutorial](https://www.youtube.com/watch?v=mTxzTWngm-Y)

[GroupBy and Aggregate Functions](https://www.youtube.com/watch?v=m4vbV27rLwA)

[Using collect\_list()](https://sparkbyexamples.com/spark/spark-collect-list-and-collect-set-functions/)

[getAs](https://spark.apache.org/docs/1.6.1/api/java/org/apache/spark/sql/Row.html#getAs%28java.lang.String%29)

[get specific row from spark dataframe](https://stackoverflow.com/questions/35243744/get-specific-row-from-spark-dataframe)

[Read Array of String from Spark](https://stackoverflow.com/questions/33204205/read-array-of-string-from-spark)

[Compare a long value is equal to Long value](https://stackoverflow.com/questions/4483734/how-to-compare-a-long-value-is-equal-to-long-value)

[Writing List to a File in Python](https://pynative.com/python-write-list-to-file/)

[How to iterate over rows in a DataFrame in Pandas](https://stackoverflow.com/questions/16476924/how-to-iterate-over-rows-in-a-dataframe-in-pandas)

[Why does "return list.sort()" return None, not the list?](https://stackoverflow.com/questions/7301110/why-does-return-list-sort-return-none-not-the-list)

[pandas.core.groupby.DataFrameGroupBy.agg](https://pandas.pydata.org/pandas-docs/version/0.22/generated/pandas.core.groupby.DataFrameGroupBy.agg.html)

[Pandas:Scaling to large datasets](https://pandas.pydata.org/docs/user\_guide/scale.html)

[How to convert a dataframe to a dictionary](https://stackoverflow.com/questions/18695605/how-to-convert-a-dataframe-to-a-dictionary)

[pandas.Series.apply](https://pandas.pydata.org/docs/reference/api/pandas.Series.apply.html#pandas-series-apply)

[pandas.DataFrame.apply](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.apply.html)