TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

BÁO CÁO BÀI TẬP LỚN MÔN HỌC

Lưu trữ và xử lý dữ liệu lớp

LƯU TRỮ, XỬ LÝ VÀ PHÂN TÍCH ĐIỂM THI ĐẠI HỌC 2020

Nhóm: **RHUST**

Phan Thành Đạt	20173001
Hoàng Văn Chương	20172984
Đỗ Minh Vũ	20173471
Nguyễn Thị Bắc	20172963

I. ĐẶT VẤN ĐỀ

Xuất phát từ nhu cầu phân tích phổ điểm, dự đoán độ tin cậy trong công tác tổ chức thi trên tập dữ liệu lớn khó thực hiện một cách thủ công.

Xuất phát từ những ứng dụng thực tiễn to lớn của lưu trữ và xử lý dữ liệu lớn.

Áp dụng các kiến thức đã học vào việc xử lý một vấn đề thực tế, liên quan trực tiếp tới môn học.

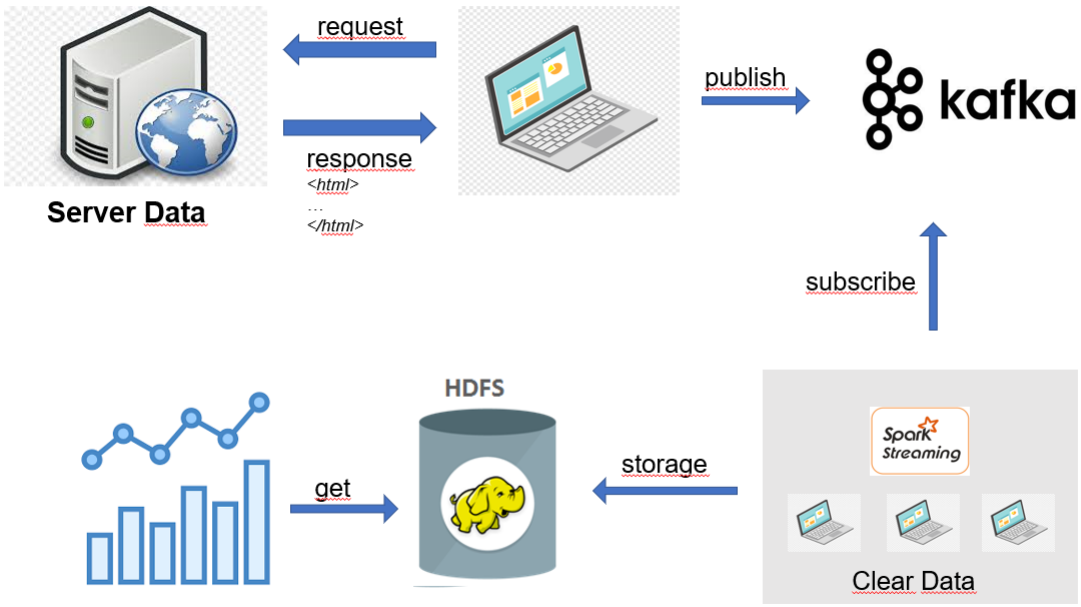
II. PHÂN TÍCH BÀI TOÁN

1. Yêu cầu

- Trích xuất thông tin về điểm thi đại học 2020.
- Làm sạch dữ liệu
- Lưu trữ dữ liêu thu được
- Truy xuất các thông tin hữu ích từ dữ liệu đã thu thập được.

2. Quy trình thực hiện

Sơ đồ:

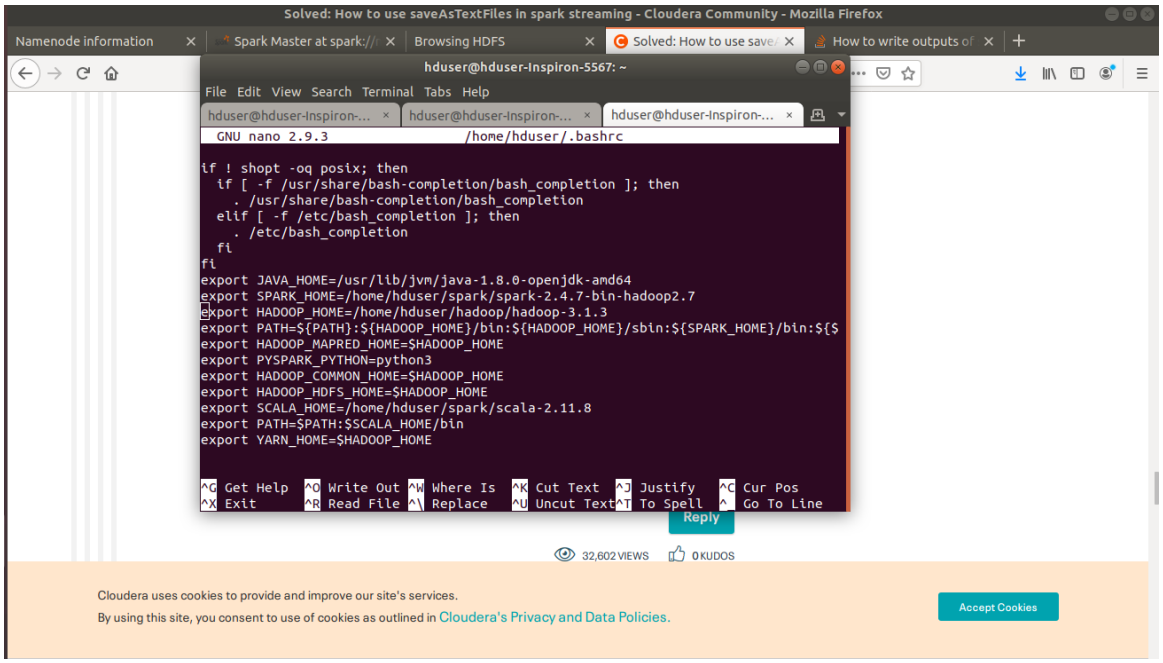


III. Triển khai

1. Cài đặt môi trường

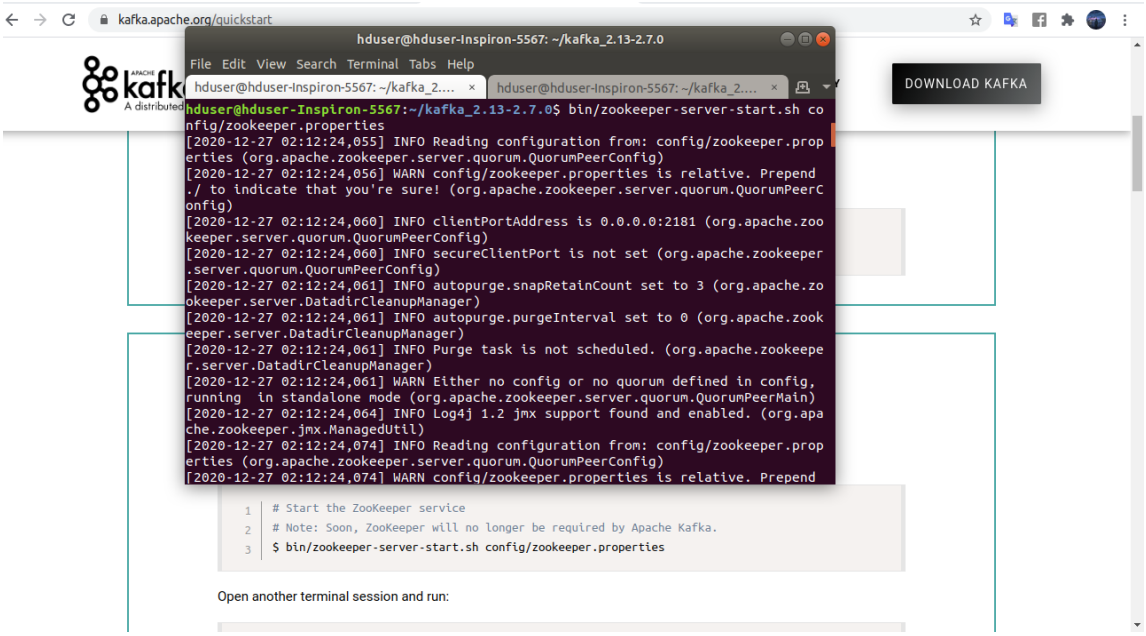
Cài đặt Java 8 và Scala 2.11.8, ssh.

Cài đặt các biến môi trường:

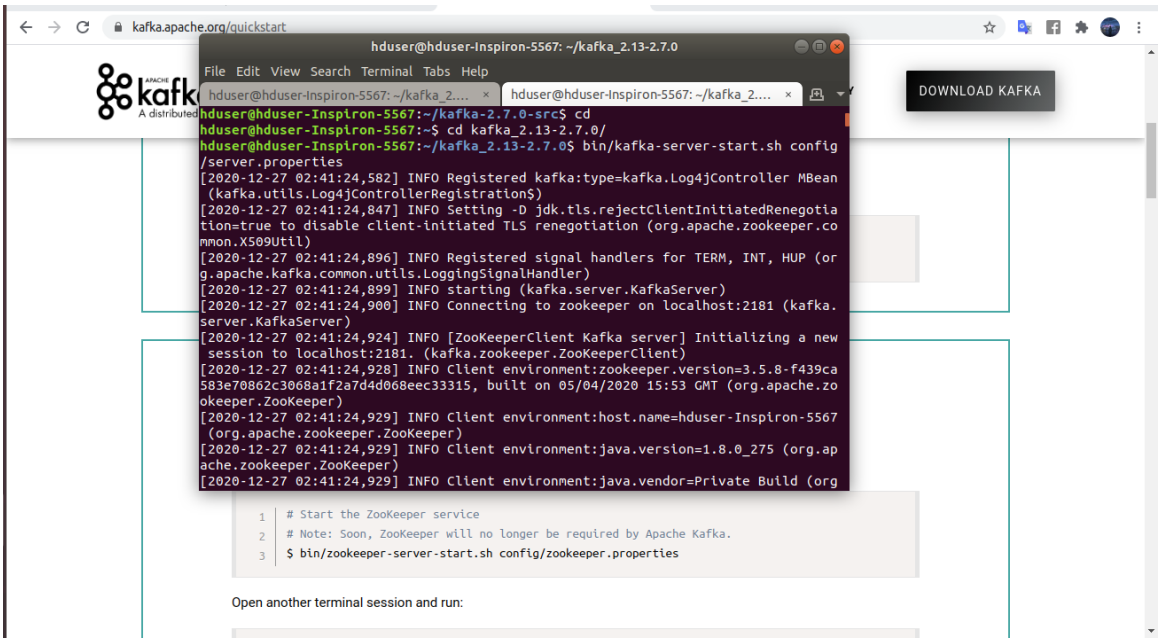


2. Cài đặt Kafka 2.13-2.7.0

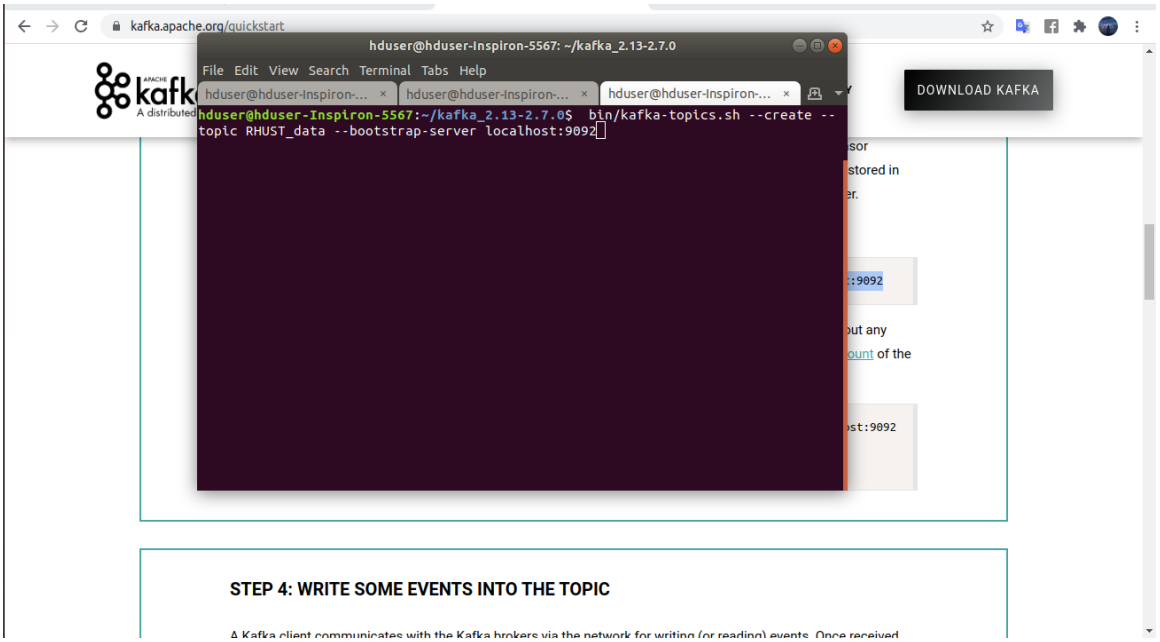
- Tải Kafka tại : https://www.apache.org/dyn/closer.cgi?path=/kafka/2.7.0/kafka_2.13-2.7.0.tgz
- Khởi tạo ZooKeeperServer : kafka/bin/zookeeper-server-start.sh config/zookeeper.properties



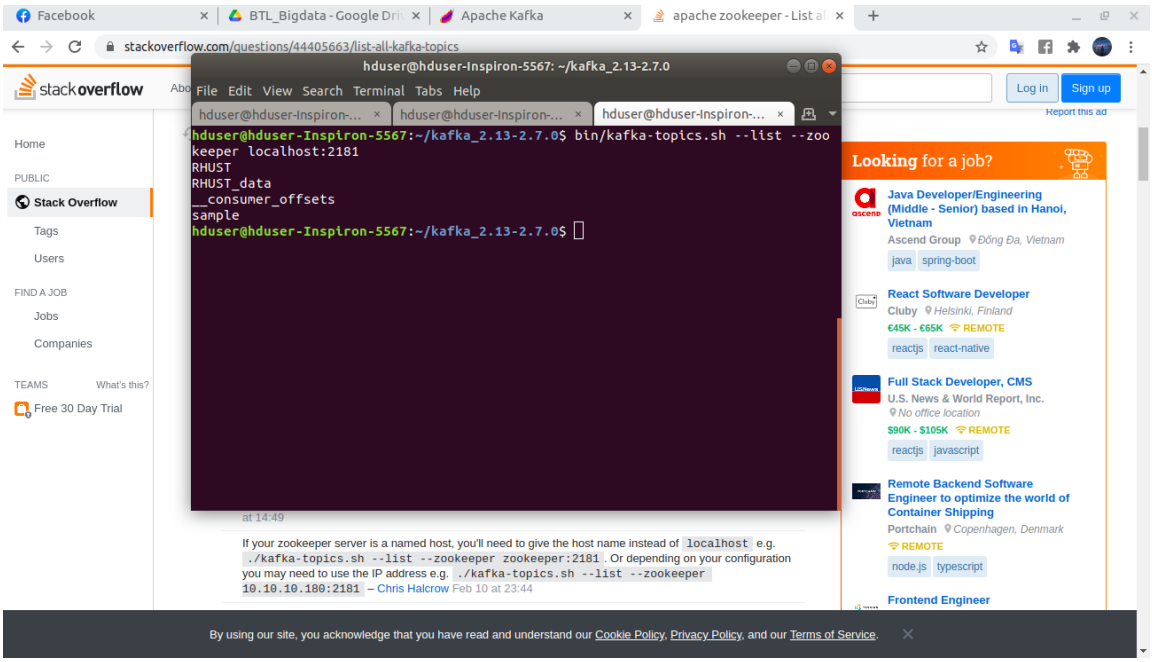
- Khởi tạo KafkaServer: kafka/bin/kafka-server-start.sh config/server.properties



- Tạo Topic trao đổi dữ liệu: bin/kafka-topics.sh --create --topic RHUST_data --bootstrap-server localhost:9092



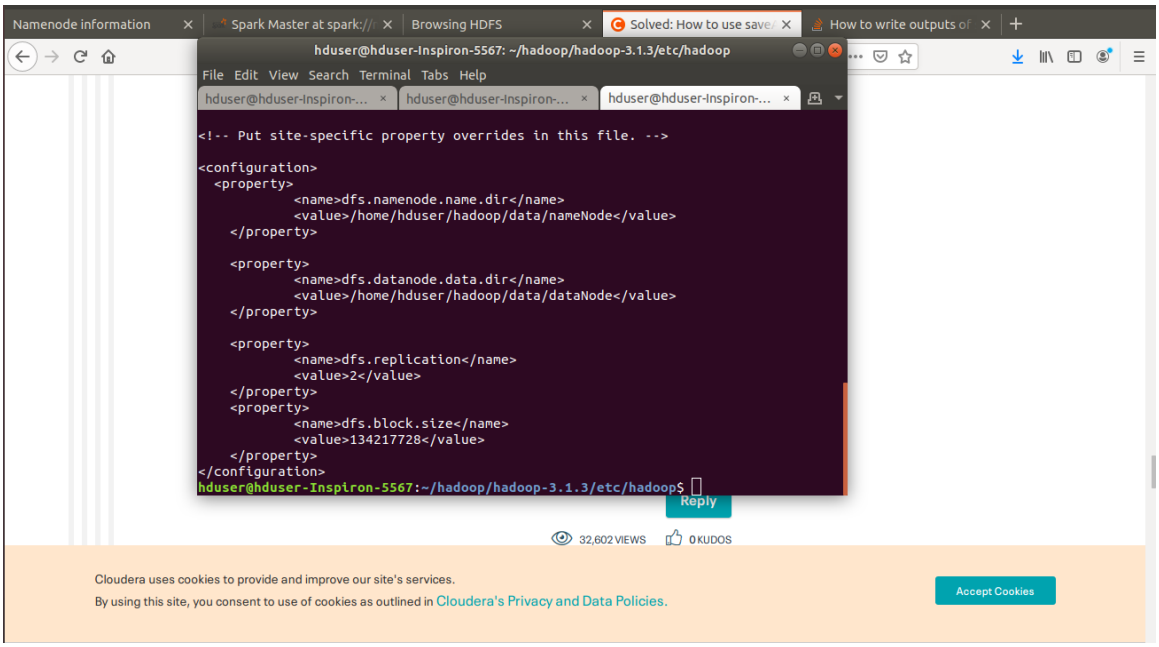
- Kiểm tra topic:

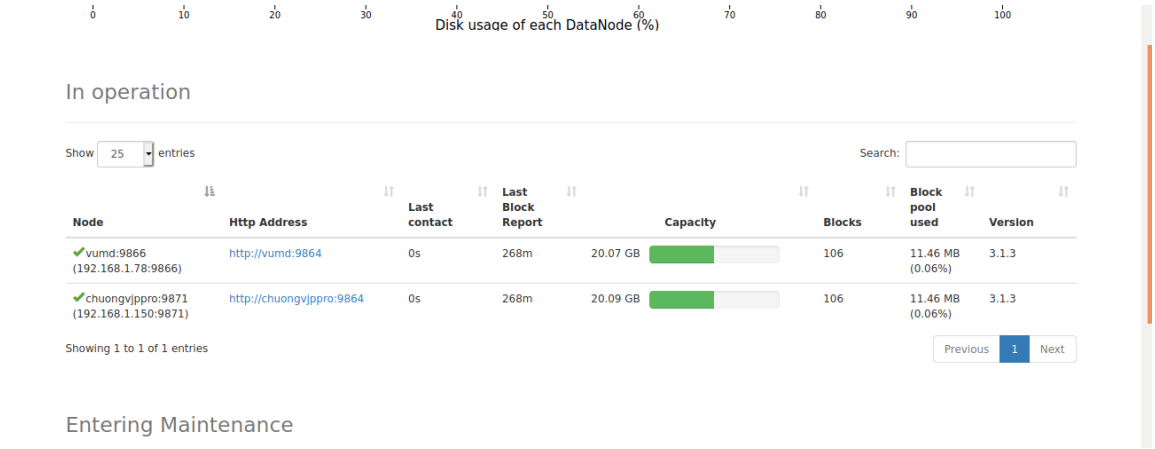


3. Cài đặt cluster HDFS

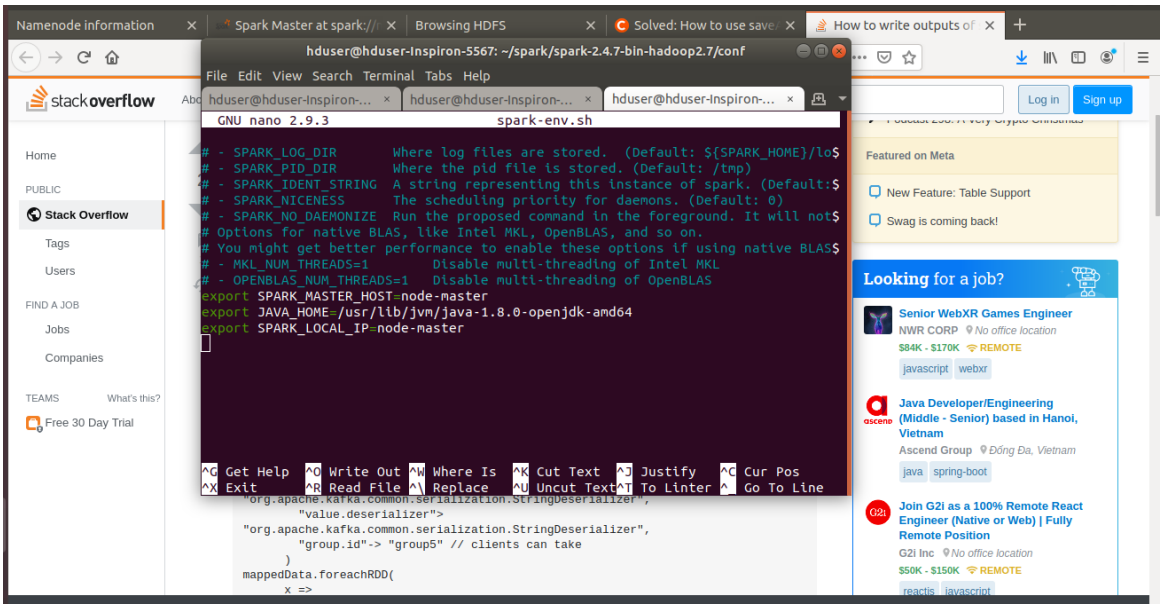
Một cụm cluster HDFS bao gồm 1 node-master và 2 node-worker.

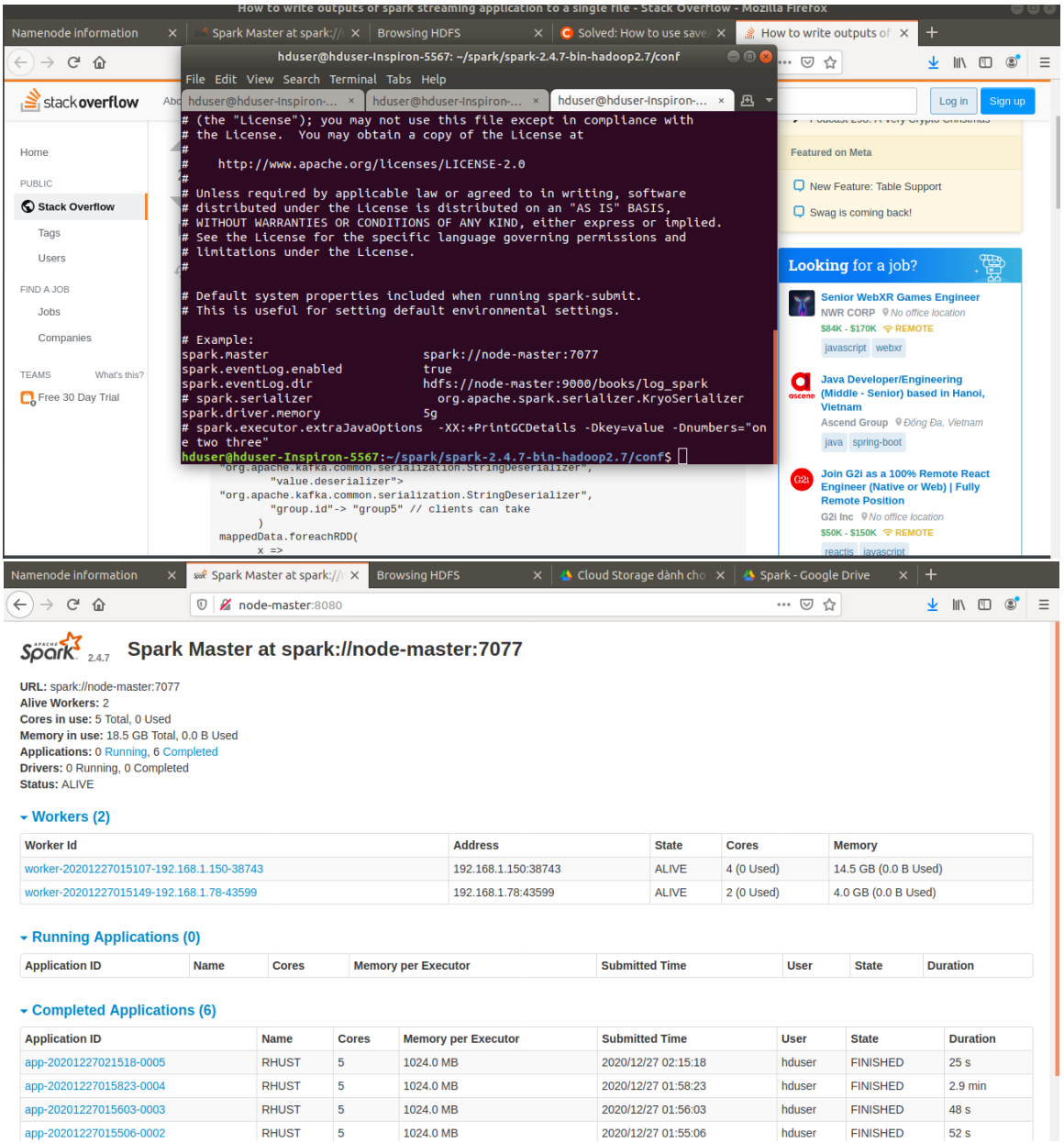
Cài đặt block-size và số lượng bản sao:





4. Cài đặt cluster Spark

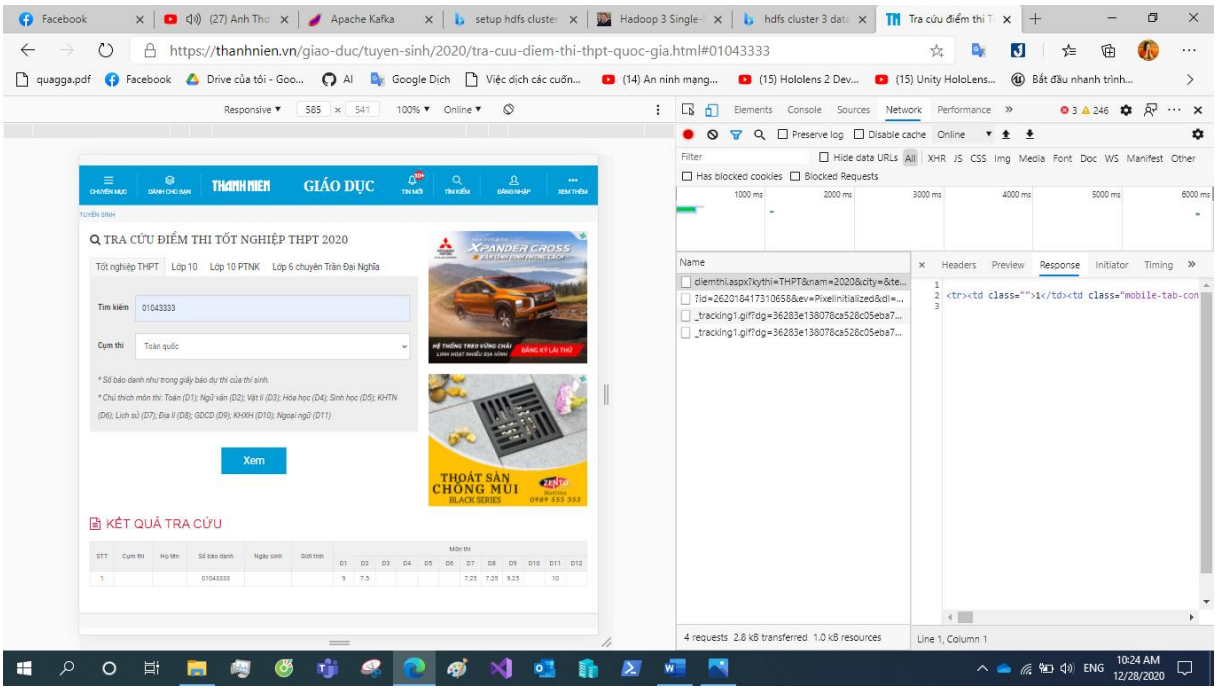




5. Lập trình

Phân tích cấu trúc trang web: https://thanhnien.vn/giao-duc/tuyen-sinh/2020/tra-cuu-diem-thi-thpt-quoc-gia.html

Mỗi khi tìm kiếm điểm theo số báo danh, 1 request sẽ gửi gửi lên và 1 response sẽ đc trả về:



URL gửi request: https://thanhnien.vn/ajax/diemthi.aspx?kythi=THPT&nam=2020&city=&text=SBD&top=no

Cấu trúc response:

<tr><td class="">1</td><td class="mobile-tab-content mobile-tab-1 visible"></td><td class="mobile-tab-content mobile-tab-1 visible"></td><td class="">01043333</td><td class="mobile-tab-content mobile-tab-1 visible"></td><td class="mobile-tab-content mobile-tab-1 visible"></td><td class="mobile-tab-content mobile-tab-2">9</td><td class="mobile-tab-content mobile-tab-2">7.5</td><td class="mobile-tab-content mobile-tab-2"></td><td class="mobile-tab-content mobile-tab-2"></td><td class="mobile-tab-content mobile-tab-2"></td><td class="mobile-tab-content mobile-tab-2"></td><td class="mobile-tab-content mobile-tab-3">7.25</td><td class="mobile-tab-content mobile-tab-3">7.25</td><td class="mobile-tab-content mobile-tab-3">9.25</td><td class="mobile-tab-content mobile-tab-3"></td><td class="mobile-tab-content mobile-tab-3">10</td><td class="mobile-tab-content mobile-tab-3"></td></tr>

Để lấy được dữ liệu thì phải tách dữ liệu từ các tag html.

Cấu trúc SBD = Mã Tỉnh +abc.def

Bắt đầu từ SBD 1

a) Source code request và gửi dữ liệu vào Kafka

```
from kafka import KafkaProducer
import requests
producer = KafkaProducer(bootstrap_servers='node-master:9092')
#SBD='02074715'
SBD='02000000'
```

```
i=1
while True:
 SBD=SBD[:len(SBD)-len(str(i))]+str(i)
 x =
requests.get('https://thanhnien.vn/ajax/diemthi.aspx?kythi=THPT&nam=2020&city=&text='+SBD+'&top=no')
 if x.text=="\n":
     break
 producer.send('RHUST_data', bytes(x.text,'utf-8'))
 producer.flush()
 i=i+1
 print(SBD)
```

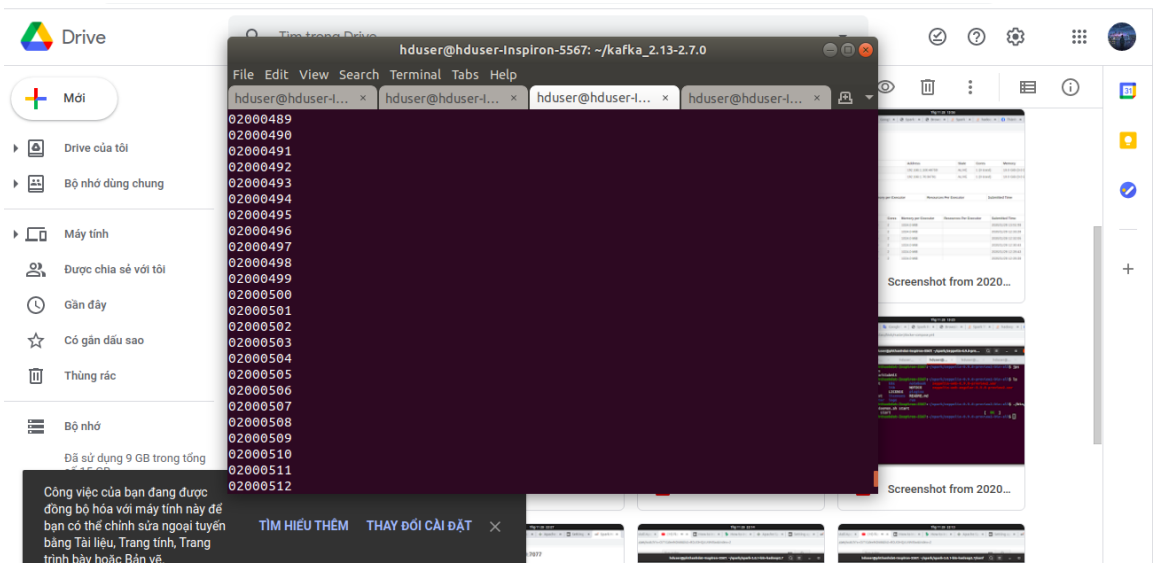
b) Source code sparkstreaming và clear data:

```
import os
from scrapy import Selector
packages = "org.apache.spark:spark-streaming-kafka-0-8_2.11:2.4.7"

os.environ["PYSPARK_SUBMIT_ARGS"] = (
    "--packages {0} pyspark-shell".format(packages)
)
from pyspark.sql.types import *
from pyspark import SparkConf
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from pyspark.streaming.kafka import KafkaUtils
from pyspark.sql.session import SparkSession
conf = SparkConf()
conf.setMaster('spark://node-master:7077')
conf.setAppName('RHUST')
sc = SparkContext(conf=conf)
sc.setLogLevel("WARN")
ss=SparkSession(sc)
ssc = StreamingContext(sc, 3600)


KAFKA_BROKER = "127.0.0.1:9092"
KAFKA_TOPIC = "RHUST_data"
# Create a schema for the dataframe
schema = StructType([
    StructField('NumCity', StringType(), True),
    StructField('MSSV', StringType(), True),
    StructField('Toan', StringType(), True),
    StructField('Van', StringType(),True),
    StructField('Ly', StringType(),True),
    StructField('HoaHoc', StringType(),True),
    StructField('Sinh', StringType(),True),
    StructField('KHTN', StringType(),True),
    StructField('Su', StringType(),True),
    StructField('Dia', StringType(),True),
    StructField('GDCD', StringType(),True),
    StructField('KHXH', StringType(),True),
    StructField('NgoaiNgu', StringType(),True),
    StructField('NotKnow', StringType(),True),
])

print("RHUST")
kafkaStream =
KafkaUtils.createDirectStream(ssc,[KAFKA_TOPIC],{"metadata.broker.list":KAFKA_BROKER})
lines=kafkaStream.map(lambda value:value[1])

def handle_rdd(value):
    result=[]
    City=Selector(text=value).xpath('//td[@class=""]/text()')[1].extract()[:2]
```

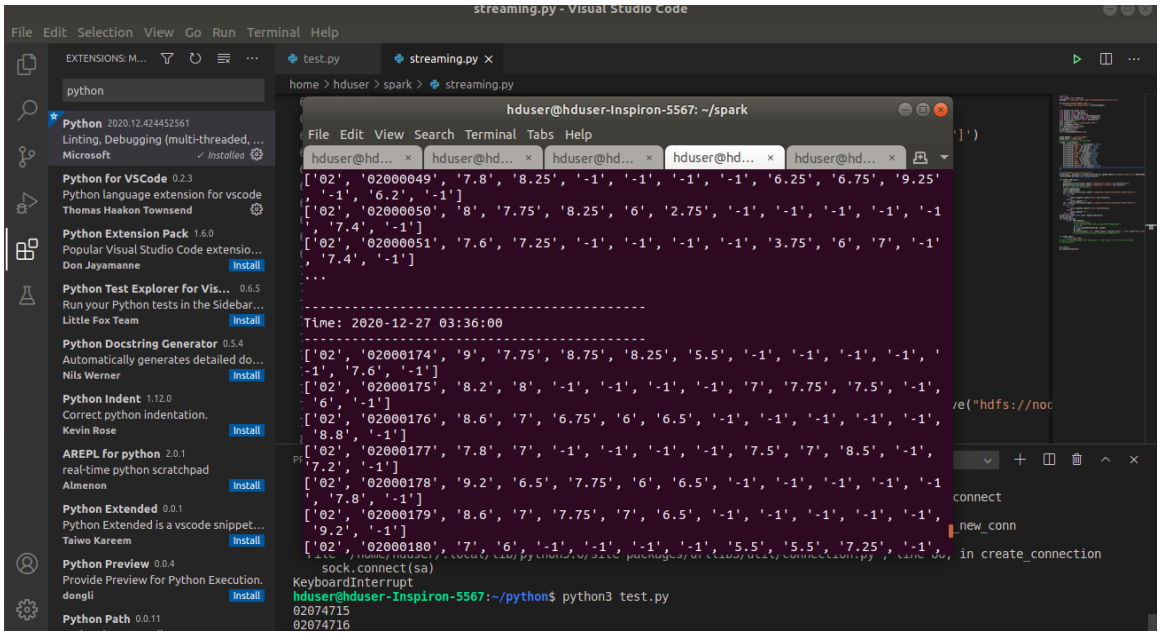
```
    MSSV=Selector(text=value).xpath('//td[@class=""]/text()')[1].extract()
    # result.insert(0,result[0][:2])
    result.append(City)
    result.append(MSSV)
    sel = Selector(text=value).xpath('//td[@class="mobile-tab-content mobile-tab-2"]')
    for i in sel:
        try:
            result.append(i.xpath("text()")[0].extract())
        except:
            result.append("-1")
    sel = Selector(text=value).xpath('//td[@class="mobile-tab-content mobile-tab-3"]')
    for i in sel:
        try:
            result.append(i.xpath("text()")[0].extract())
        except:
            result.append("-1")
    return result
coords=lines.map(lambda value: handle_rdd(value))
def store_rdd(rdd):
        if not rdd.isEmpty():
                print(type(rdd))
                global ss
                df = ss.createDataFrame(rdd, schema)
                df.write.format('csv').mode('append').option("header",
"true").csv("hdfs://node-master:9000/data_thpt")
                df.show()
               # rdd.saveAsTextFile("hdfs://node-master:9000/books")

def empty_rdd():
        print("empty RDD")
coords.foreachRDD(lambda rdd: empty_rdd() if rdd.count() == 0 else store_rdd(rdd))
# coords.pprint()
ssc.start()
ssc.awaitTermination()
```

6. Thực thi source code

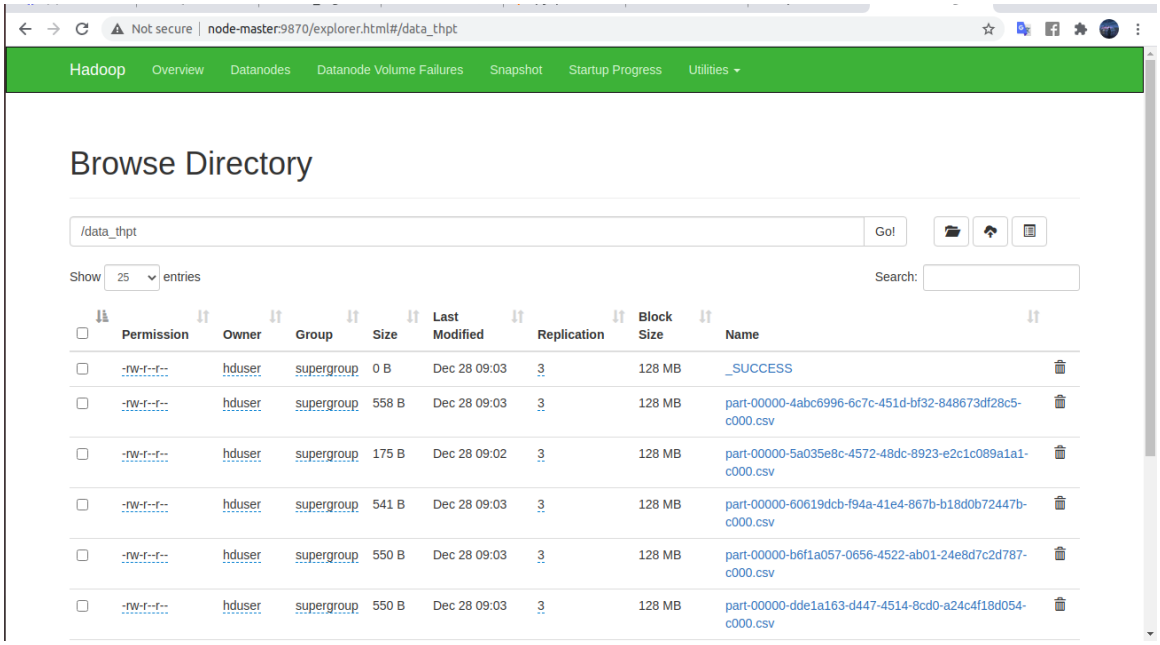
Chạy python3 produce.py để publish dữ liệu vào kafka



Chạy spark-submit --packages org.apache.spark:spark-streaming-kafka-0-8_2.11:2.4.7 streaming.py để chạy streaming



Kiểm tra output:

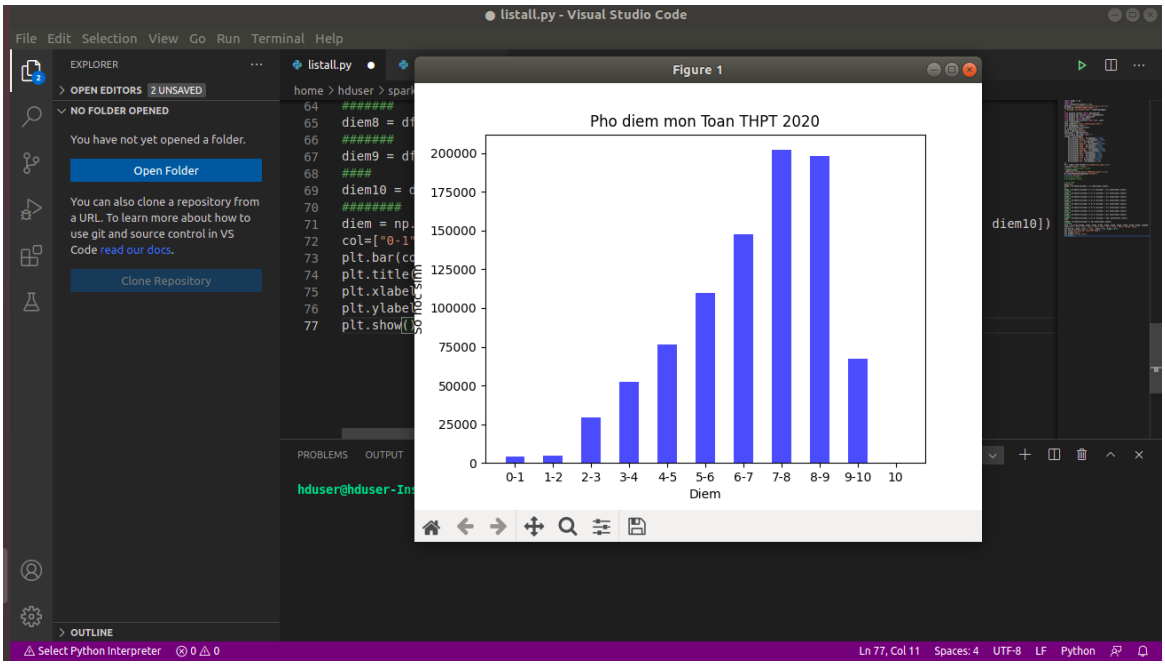


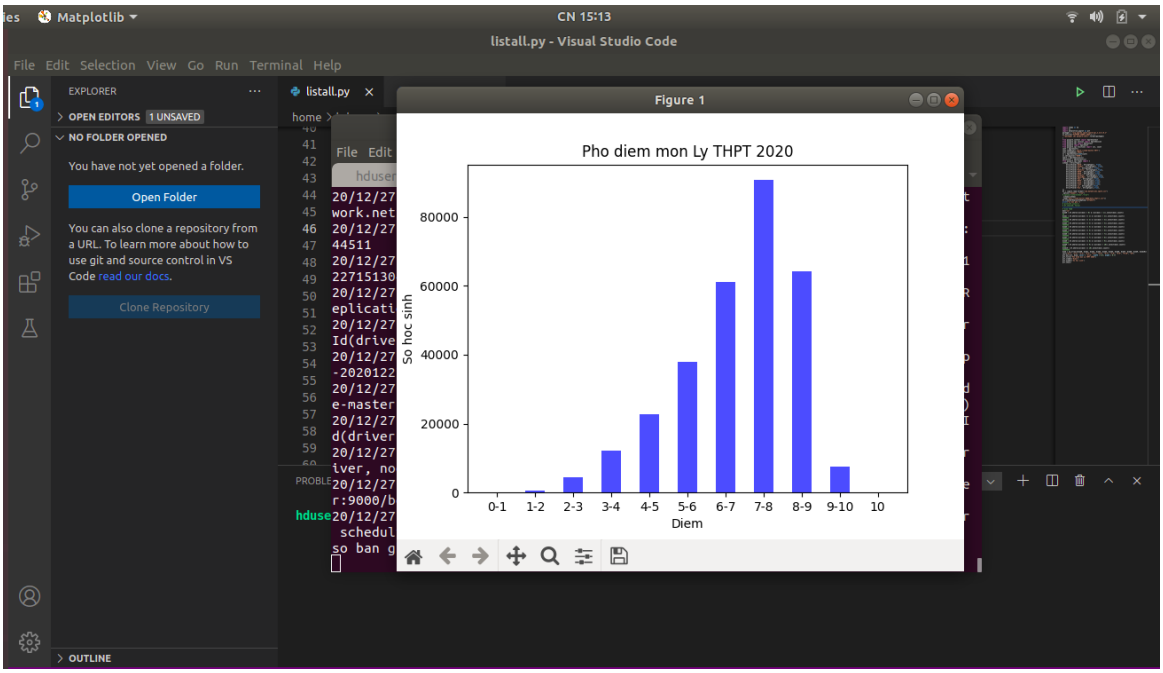
7. Trực quan hóa dữ liệu

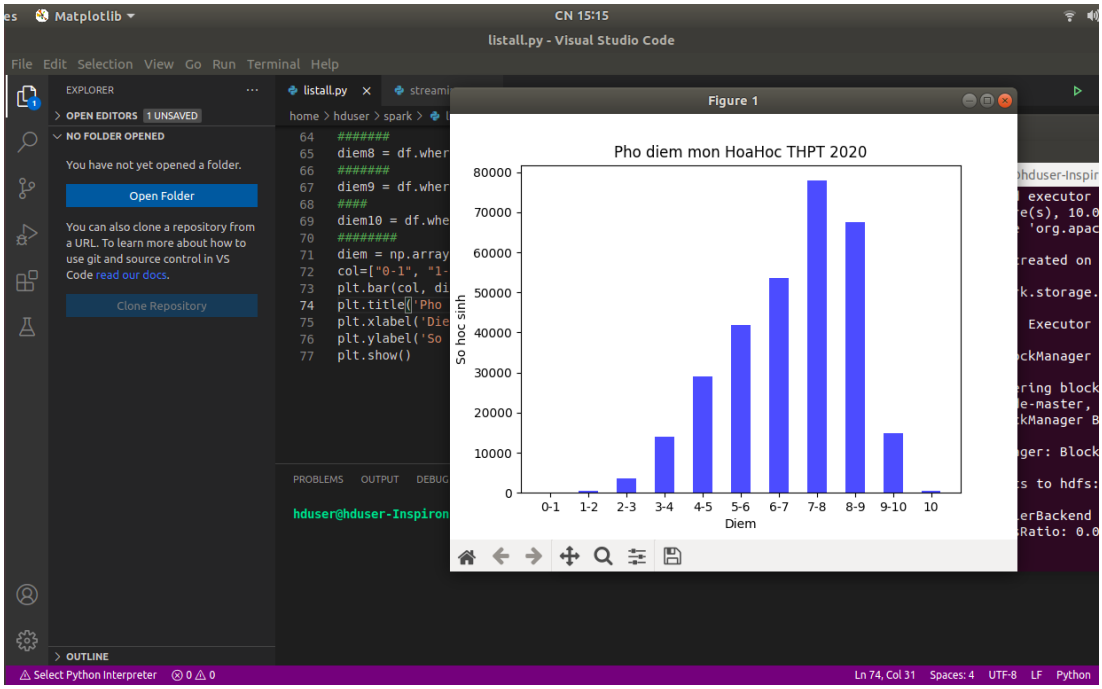
```
import numpy as np
import os
import matplotlib.pyplot as plt
packages = "org.apache.spark:spark-sql_2.12:3.0.1"
os.environ["PYSPARK_SUBMIT_ARGS"] = (
"--packages {0} pyspark-shell".format(packages)
)
from pyspark.context import SparkContext
from pyspark.sql.session import SparkSession
from pyspark.sql import SQLContext
from pyspark import SparkConf
from pyspark.sql.functions import *
import pyspark.sql.functions as psf
conf = SparkConf()
conf.setMaster('spark://node-master:7077')
```

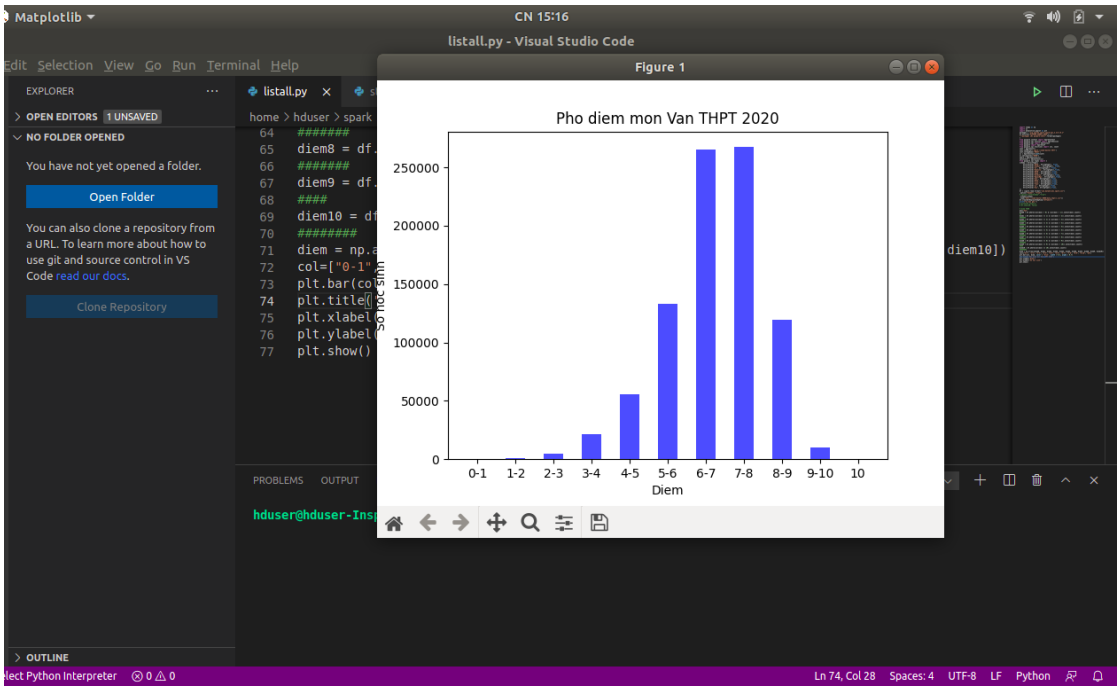
```
conf.setAppName('RHUST')
sc = SparkContext(conf=conf)
sc.setLogLevel("WARN")
spark = SparkSession(sc)
sqlContext = SQLContext(sc)
from pyspark.sql.types import *
from pyspark.sql.functions import sum
schema = StructType([
    StructField('GDCD', StringType(), True),
    StructField('HoaHoc', StringType(), True),
    StructField('Su', StringType(), True),
    StructField('NumCity', StringType(),True),
    StructField('KHTN', StringType(),True),
    StructField('KHXH', StringType(),True),
    StructField('NotKnow', StringType(),True),
    StructField('NgoaiNgu', StringType(),True),
    StructField('Van', StringType(),True),
    StructField('MSSV', StringType(),True),
    StructField('Sinh', StringType(),True),
    StructField('Toan', StringType(),True),
    StructField('Ly', StringType(),True),
    StructField('Dia', StringType(),True),
])
df = (spark.read.format("com.databricks.spark.csv")
.option("header", "true")
#.option("inferSchema","true")
.schema(schema)
.load("hdfs://node-master:9000/data_thpt/*.csv"))
df.createOrReplaceTempView("dfTable")
print("so ban ghi:")
# print(df.count())
# df.show(20, False)

#diem TOAN
# mon="Van"
# diem0 = df.where((col(mon) > 0) & (col(mon) < 1)).select(mon).count()
# ##
# diem1 = df.where((col(mon) >= 1) & (col(mon) < 2)).select(mon).count()
# #######
# diem2 = df.where((col(mon) >= 2) & (col(mon) < 3)).select(mon).count()
# #######
# diem3 = df.where((col(mon) >= 3) & (col(mon) < 4)).select(mon).count()
# #######
# diem4 = df.where((col(mon) >= 4) & (col(mon) < 5)).select(mon).count()
# #######
# diem5 = df.where((col(mon) >= 5) & (col(mon) < 6)).select(mon).count()
# #######
# diem6 = df.where((col(mon) >= 6) & (col(mon) < 7)).select(mon).count()
# #######
# diem7 = df.where((col(mon) >= 7) & (col(mon) < 8)).select(mon).count()
# #######
# diem8 = df.where((col(mon) >= 8) & (col(mon) < 9)).select(mon).count()
# #######
# diem9 = df.where((col(mon) >= 9) & (col(mon) < 10)).select(mon).count()
# ####
# diem10 = df.where(col(mon) == 10).select(mon).count()

########
# diem = np.array([diem0, diem1, diem2, diem3, diem4, diem5, diem6, diem7,
diem8, diem9, diem10])
# col=["0-1", "1-2", "2-3", "3-4", "4-5", "5-6", "6-7", "7-8", "8-9", "9-10",
"10"]
# plt.bar(col, diem, color = 'blue', width = 0.5, alpha = 0.7)
# plt.title('Pho diem mon Van THPT 2020')
# plt.xlabel('Diem')
# plt.ylabel('So hoc sinh')
# plt.show()
other=df.select().count()
```

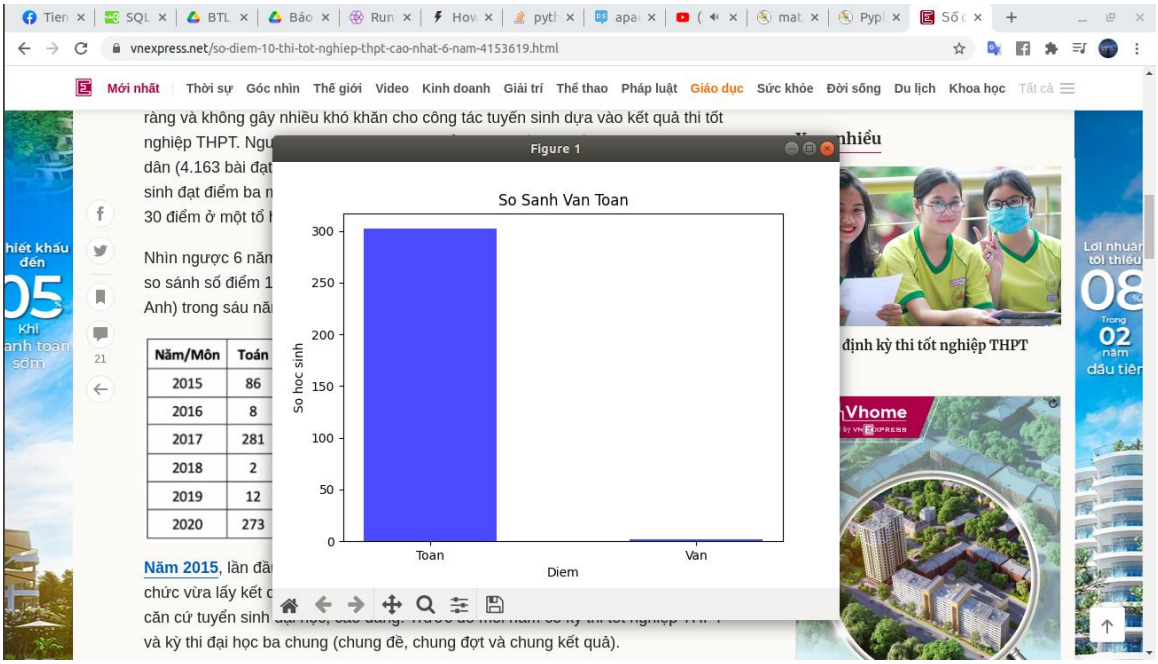
```
KHTN = df.where((col("Sinh") != -1) & (col("Ly") != -1) & (col("HoaHoc") != -
1)).select().count()
KHXH = df.where((col("GDCD") != -1) & (col("Su") != -1) & (col("Dia") != -
1)).select().count()
print(other)
print(KHTN)
print(KHXH)
diem = np.array([KHTN, KHXH])
label=["KHTN", "KHXH"]
plt.pie(diem, labels=label, startangle = 90, autopct='%.1f%%')
plt.show()
# Toan = df.where(col("Toan")==10).select().count()
# Van = df.where(col("Van") == 10).select().count()
# diem = np.array([Toan, Van])
# col=["Toan","Van"]
# plt.bar(col, diem, color = 'blue', width = 0.5, alpha = 0.7)
# plt.title('So Sanh Van Toan')
# plt.xlabel('Diem')
# plt.ylabel('So hoc sinh')
# plt.show()


# Toan = df.where(col("Toan")==10).select().count()
# GDCD = df.where(col("GDCD") == 10).select().count()
# HoaHoc = df.where(col("HoaHoc") == 10).select().count()
# NgoaiNgu = df.where(col("NgoaiNgu") == 10).select().count()
# Sinh = df.where(col("Sinh") == 10).select().count()
# Ly = df.where(col("Ly") == 10).select().count()
# Dia = df.where(col("Dia") == 10).select().count()
# Su = df.where(col("Su") == 10).select().count()
# Van = df.where(col("Van") == 10).select().count()
# diem = np.array([Toan, Van, GDCD, HoaHoc, NgoaiNgu, Sinh, Ly, Dia, Su])
# label=["Toan", "Van", "GDCD", "HoaHoc", "NgoaiNgu", "Sinh", "Ly", "Dia", "Su"]
# plt.pie(diem, labels=label, startangle = 90, autopct='%.1f%%')
# plt.show()
```

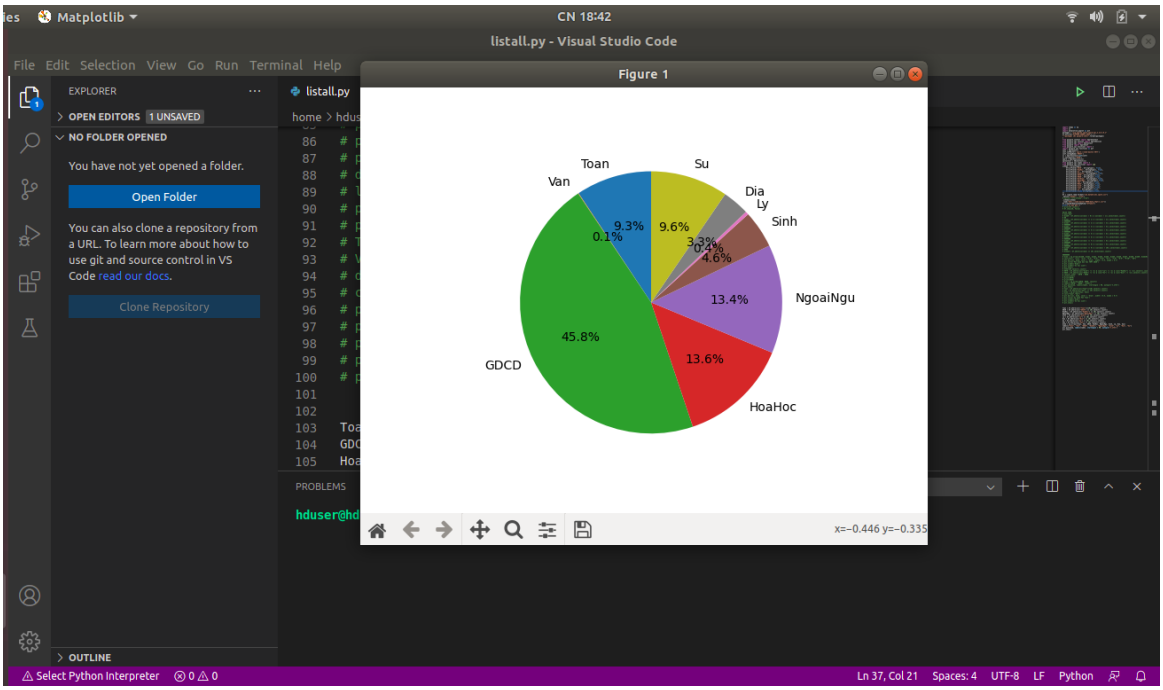


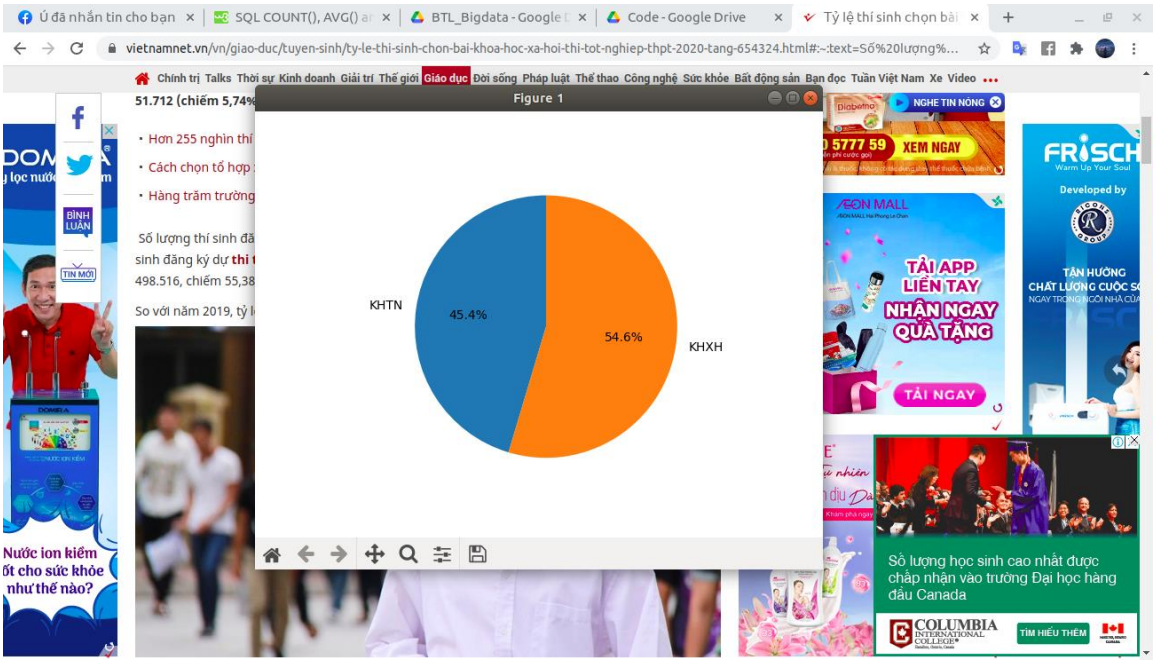

Matplotlib
CN 15:13
listall.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
You can also clone a repository from a URL. To learn more about how to use git and source control in VS Code read our docs.
Clone Repository
OUTLINE
listall.py
Figure 1
Pho diem mon Ly THPT 2020
So hoc sinh
Diem
0-1
1-2
2-3
3-4
4-5
5-6
6-7
7-8
8-9
9-10
10
0
20000
40000
60000
80000


Matplotlib
CN 15:15
listall.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
You can also clone a repository from a URL. To learn more about how to use git and source control in VS Code read our docs.
Clone Repository
OUTLINE
PROBLEMS
OUTPUT
hduser@hduser-Inspiron
Figure 1
Pho diem mon HoaHoc THPT 2020
So hoc sinh
Diem
0-1
1-2
2-3
3-4
4-5
5-6
6-7
7-8
8-9
9-10
10
0
10000
20000
30000
40000
50000
60000
70000
80000
Select Python Interpreter
Ln 74, Col 31
Spaces: 4
UTF-8
LF
Python


Matplotlib
CN 15:16
listall.py - Visual Studio Code
Figure 1
Pho diem mon Van THPT 2020
So hoc sinh
Diem
0-1
1-2
2-3
3-4
4-5
5-6
6-7
7-8
8-9
9-10
10
0
50000
100000
150000
200000
250000
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Clone Repository
OUTLINE


vnexpress.net/so-diem-10-thi-tot-nghiep-thpt-cao-nhat-6-nam-4153619.html
Figure 1
So Sanh Van Toan
So hoc sinh
Diem
Toan
Van
0
50
100
150
200
250
300
Năm/Môn
Toán
2015
86
2016
8
2017
281
2018
2
2019
12
2020
273
định kỳ thi tốt nghiệp THPT





Source full: https://github.com/phanthanhdat1902/SparkStreamingKafka

8. Source code scrapy cho từng máy

```
import scrapy
from thpt.items import ThptItem


class CrawlerSpider(scrapy.Spider):
    name = "crawl_point"
    MaVung="02"
    start_urls = [
        'https://thanhnien.vn/giao-duc/tuyen-sinh/2020/tra-cuu-diem-thi-thpt-quoc-gia.html'
    ]

    def parse(self, response):
        SBD='02000000'
        for i in range(201,300):
```

```
            SBD=SBD[:len(SBD)-len(str(i))]+str(i)
            print(i)
            yield
scrapy.Request("https://thanhnien.vn/ajax/diemthi.aspx?kythi=THPT&nam=2020&city=&text="+SBD+"&top=no",
callback=self.crawlLyric)

    def crawlLyric(self, response):
        item = ThptItem()
        print(response.xpath("//td[@class='']/text()")[1].extract())
        item['SBD']=response.xpath("//td[@class='']/text()")[1].extract()
        item['MaVung']=item['SBD'][0:2]
        tab_2=["Toán","Ngữvăn","Vậtlí","Hóahọc","Sinhhọc"]
        tab_2=list(enumerate(tab_2))
        tab_3=["Lịchsử","Địalí","GDCD","Ngoạingữ","N1"]
        tab_3=list(enumerate(tab_3))
        #set default NgoaiNgu
        item['N2'] = -1
        item['N3'] = -1
        item['N4'] = -1
        item['N5'] = -1
        item['N6'] = -1
        arrTr = response.xpath("//td[@class='mobile-tab-content mobile-tab-2']")
        for i in tab_2:
           try:
              item[i[1]]=arrTr[i[0]].xpath("text()")[0].extract()
           except:
              item[i[1]]=-1
        arrTr = response.xpath("//td[@class='mobile-tab-content mobile-tab-2']")
        for i in tab_3:
           try:
              item[i[1]]=arrTr[i[0]].xpath("text()")[0].extract()
           except:
              item[i[1]]=-1
        item['N1'] = item["Ngoạingữ"]
        yield item
        # print(response.xpath('//table[@class="table thpt-mobile hidden-md hidden-sm hidden-
lg"]/tr')[0].xpath('td/text()').extract())
```

File item:

```
# Define here the models for your scraped items
#
# See documentation in:
# https://docs.scrapy.org/en/latest/topics/items.html

import scrapy

class ThptItem(scrapy.Item):
    # define the fields for your item here like:
    # name = scrapy.Field()
    MaVung=scrapy.Field()
    SBD=scrapy.Field()
    Toán = scrapy.Field()
    Ngữvăn = scrapy.Field()
    Ngoạingữ=scrapy.Field()
    N1 = scrapy.Field()
    N2 = scrapy.Field()
    N3 = scrapy.Field()
    N4 = scrapy.Field()
    N5 = scrapy.Field()
    N6=scrapy.Field()
    Vậtlí=scrapy.Field()
    Hóahọc =scrapy.Field()
    Sinhhọc=scrapy.Field()
    Lịchsử=scrapy.Field()
    Địalí=scrapy.Field()
    GDCD=scrapy.Field()
    pass
```

IV. Hướng phát triển tương lai

Phát triển hệ thống dự đoán điểm chuẩn đại học, dự đoán khả năng gian lận trong thi cử.