AI VIET NAM – COURSE 2023

Big Data Exercises

(Hadoop and its Application)

Ngày 6 tháng 8 năm 2023

Câu 1: Dữ liệu có kích thước ___ byte được gọi là Big Data:

(A) Meta

(B) Peta

(C) Giga

(D) Tera

Câu 2: Hãy liệt kê danh sách big data's 5v:

(A) Volume, Velocity, Variety, Value and Veracity

(B) Volume, Velocity, Variable, Value and Veracity

(C) Volume, Velocity, Validity, Value and Veracity

(D) Volume, Vulnerability, Variety, Value and Veracity

Câu 3: Theo bạn khái niệm Velocity đề cập trong big data's 5v có ý nghĩa gì:

(A) Data can arrive at fast speed

(B) Enormous datasets can accumulate within very short periods of time

(C) Velocity of data translates into the amount of time it takes for the data to be processed

(D) cả 3 (A), (B), (C)

Câu 4: Trong số những công nghệ sau đây, công nghệ nào không được dùng cho Big Data?

(A) Apache Hadoop

(B) Apache Spark

(C) Apache Pytarch

(D) Apache Kafka

Câu 5: Hãy chọn đáp án đúng cho dữ liệu phi cấu trúc bên dưới (unstructured data)?

(A) Students roll number, age

(B) Videos

(C) Audio files

(D) cả B and C

Câu 6: Định nghĩa nào sau đây là đúng về Hadoop Ecosystem?

(A) Hive là một cơ sở dữ liệu quan hệ hỗ trợ các truy vấn SQL.

(B) Pig là một cơ sở dữ liệu quan hệ hỗ trợ truy vấn SQL.

(C) Cả A và B

(D) Không câu nào đúng

Câu 7: Hướng dẫn cài đặt và sử dụng Hadoop trên Google Colab

a. Cài đặt các thư viện cơ bản:

```

# install java
!apt-get install openjdk-8-jdk-headless -qq > /dev/null

#create java home variable
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"

#extract the file
!tar -xzvf hadoop-3.3.0.tar.gz


```

b. Tải và giải nén hadoop-3.3.0.tar.gz:

```

#download HADOOP (NEW DOWNLOAD LINK)
!wget https://archive.apache.org/dist/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz

#extract the file
!tar -xzvf hadoop-3.3.0.tar.gz


```

c. Cấu hình sử dụng Hadoop:

```

#copy the hadoop file to user/local
!cp -r hadoop-3.3.0/ /usr/local/

#find  the default Java path
!readlink -f /usr/bin/java | sed "s:bin/java::"

#run Hadoop
!/usr/local/hadoop-3.3.0/bin/hadoop

```

d. Chạy ví dụ MapReduce. Hình 1 thể hiện kết quả xử lý sau khi thực hiện MapReduce process:

```
2023-08-04 04:40:57,933 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2023-08-04 04:40:58,026 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2023-08-04 04:40:58,027 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2023-08-04 04:40:58,197 INFO input.FileInputFormat: Total input files to process : 10
2023-08-04 04:40:58,226 INFO mapreduce.JobSubmitter: number of splits:10
2023-08-04 04:40:58,396 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local733847140_0001
2023-08-04 04:40:58,396 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-08-04 04:40:58,572 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2023-08-04 04:40:58,573 INFO mapreduce.Job: Running job: job_local733847140_0001
2023-08-04 04:40:58,578 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2023-08-04 04:40:58,589 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
```

Hình 1: Kết quả thực hiện MapReduce trên 10 files sử dụng Hadoop

```
1
2 #create input folder for demonstration exercise
3 !mkdir /content/cau7
4
5 #copy sample files to the input folder
6 !cp /usr/local/hadoop-3.3.0/etc/hadoop/*.xml /content/cau7
7
8 #check that files have been successfully copied to the input folder
9 !ls /content/cau7
10
11 #run the mapreduce example #run the mapreduce example program
12 !/usr/local/hadoop-3.3.0/bin/hadoop jar /usr/local/hadoop-3.3.0/share/hadoop/mapreduce
    /hadoop-mapreduce-examples-3.3.0.jar grep /content/drive/MyDrive/AI2023/
    Hotel_Reviews.csv /content/ket_qua 'allowed[.]*'
13
14
15
```

Câu 8: Cho trước 2 file dữ liệu **file_1.txt** và **file_2.txt** chứa thông tin về tên các hội nghị và tạp chí đầu ngành trong lĩnh vực trí tuệ nhân tạo (Hình 2). Hãy phát triển chương trình đếm tổng số lần xuất hiện của từng word trong 2 file dữ liệu trên sử dụng thư viện numpy.

Để hoàn thiện chương trình trên, bạn cần lần lượt implement các class và function sau đây:

a. Tạo file mapper.py và implement các bước trong quy trình Map của Hadooop như sau :

```
1
2 #It will read data from *STDIN, split it into words and output a list of lines mapping
     words to their counts to *STDOUT.
3
4 import sys
5 import io
6 import re
7 import nltk
8 punctuations = '''!()-[]{};:'"\,<>./?@#$%^&*_~'''
9
10 input_stream = io.TextIOWrapper(sys.stdin.buffer, encoding='latin1')
11 for line in input_stream:
12   line = line.strip()
13   line = re.sub(r'[^\w\s]', '',line)
14   line = line.lower()
15   for x in line:
16     if x in punctuations:
17       line=line.replace(x, " ")
18
```

```
  words=line.split()
  for word in words:
    print('%s\t%s' % (word, 1))


```

b. Tạo file reducer.py và implement các bước cần thiết trong quy trình Map của Hadooop như sau :

```
#It will read the results of mapper.py from STDIN and sum the occurrences of each word
     to a final count, and then output its results to STDOUT.


    from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    line=line.lower()

    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)
    try:
      count = int(count)
    except ValueError:
      #count was not a number, so silently
      #ignore/discard this line
      continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to STDOUT
            print ('%s\t%s' % (current_word, current_count))
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
    print( '%s\t%s' % (current_word, current_count))


```

:

c. Gán quyền thực thi cho 2 file mapper.py và reducer.py như sau :

```

!chmod u+rwx /content/mapper.py
!chmod u+rwx /content/reducer.py

```

d. Thực hiện đoạn chương trình sau để tính số lần xuất hiện của từng word :

```
!cat /content/cau7/ket_qua/* | python mapper.py | sort | python reducer.py
```

Hãy cho biết kết quả khi thực hiện chương trình trên

(A) *workshops* xuất hiện 1, *vision* xuất hiện 4 lần

(B) *transactions* xuất hiện 4, *society* xuất hiện 4 lần

(C) *processing* xuất hiện 2, *pattern* xuất hiện 3 lần

(D) (A), (B) và (C) đều sai

e. Sử dụng hadoop để hiện thực MapReduce bằng thư viện hadoop-streaming-3.3.0.jar. Hình 3 thể hiện kết quả bằng cách sử dụng Hadoop :

```
!/usr/local/hadoop-3.3.0/bin/hadoop jar /usr/local/hadoop-3.3.0/share/hadoop/tools/lib
    /hadoop-streaming-3.3.0.jar -input /content/drive/MyDrive/AI2023/dataset_hadoop/* -
    output /content/result7 -file /content/mapper.py  -file /content/reducer.py  -
    mapper 'python mapper.py'  -reducer 'python reducer.py'
```

Hãy cho biết kết quả khi thực hiện chương trình trên

(A) *analysis* xuất hiện 1, *computer* xuất hiện 5 lần

(B) *and* xuất hiện 3, *conference* xuất hiện 4 lần

(C) *ieee* xuất hiện 2, *ieeecvf* xuất hiện 3 lần

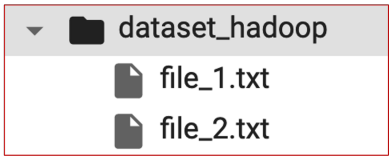
(D) (A), (B) và (C) đều đúng

file_1.txt

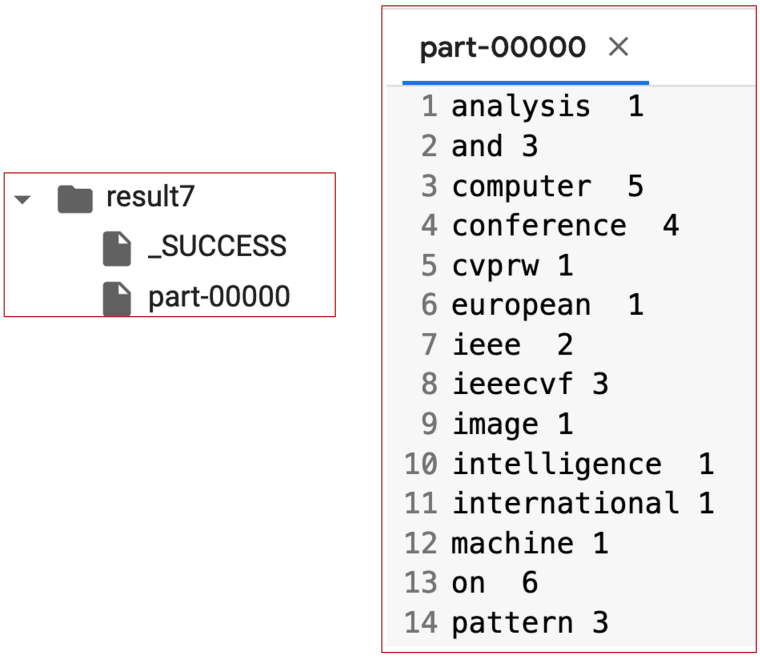
```
1 IEEE/CVF Conference on Computer Vision and Pattern Recognition
2 European Conference on Computer Vision
3 IEEE/CVF International Conference on Computer Vision
```

file_2.txt

```
1 IEEE Transactions on Pattern Analysis and Machine Intelligence
2 IEEE Transactions on Image Processing
3 IEEE/CVF Computer Society Conference on Computer Vision and Pattern
```



Hình 2: Dữ liệu để thực hiện 2 thao tác Map và Reduce để tính tổng số lần xuất hiện của từng word.



Hình 3: Kết quả MapReduce sử dụng Hadoop.