1. Bản chất DStream:

   a, là một chuỗi liên tục RDD

   b, Là một chuỗi liên tục DataFrame

   c, Là một chuỗi liên tục DataSet

   d, ko có đáp án đúng

2. Trong hệ sinh thái của Spark không có công cụ hay thành phần nào sau đây:
   a, MLib
   b, GraphX
   c, Sqoop
   d, Cluster Managers

3. Hadoop giải quyết bài toán khả mở bằng cách nào? Chọn đáp án sai
   a, Thiết kế phân tán ngay từ đầu, mặc định triển khai trên cụm máy chủ
   b, Các node tham gia vào cụm Hadoop được gán vai trò hoặc là node tính toán hoặc là node lưu trữ dữ liệu
   c, Các node tham gia vào cụm đóng cả 2 vai trò tính toán và lưu trữ
   d, Các node thêm vào cụm có thể có cấu hình, độ tin cậy cao

4. Hadoop giải quyết bài toán chịu lỗi thông qua kỹ thuật gì? Chọn đáp án **SAI**
   a, Kỹ thuật dư thừa
   b, Các tệp tin được phân mảnh, các mảnh được nhân bản ra các node khác trên cụm
   c, Các tệp tin được phân mảnh, các mảnh được lưu trữ tin cậy trên ổ cứng theo cơ chế RAID
   d, các công việc cần tính toán được phân mảnh thành các tác vụ độc lập

   c

5. Các đặc trưng của HDFS. Chọn đáp án SAI
   a, Tối ưu cho các tệp tin có kích thước lớn
   b, Hỗ trợ thao tác đọc ghi tương tranh tại chunk (phân mảnh) trên tệp tin
   c, Hỗ trợ nén dữ liệu để tiết kiệm chi phí
   d, hỗ trợ cơ chế phân quyền và kiểm soát người dùng của UNIX

6. Spark Streaming trừu tượng hóa cũng như thao tác với các dòng dữ liệu (data stream) dựa trên khái niệm nào:
   a, shared variable
   b, RDD
   c, DStream
   d, DataFrame

7. Spark hỗ trợ các cluster manager nào
   a, Standalone Cluster manager
   b, MESOS
   c, YARN
   d, tất cả đáp án trên

8. Đáp án nào không phải là một "output operation" khi thao tác với DStream
   a, saveAsTextFile
   b, foreachRDD
   c, saveasHadoopFile
   d, reduceByKeyAndWindow

9. Đáp án nào không phải là một "Transformation" khi thao tác với DStream
   a, reduceByWindow
   b, window
   c, foreachWindow
   d, countByWindow

10. Mục đích của sử dụng sparkML là gì
    a, chạy MapReduce
    b, chạy các thuật toán dự đoán
    c, tính toán phân tán
    d, cả b và c

11. Mục đích của lệnh sau đây là gì:
    (trainingData, testData) = dataset.randomSplit([0.8, 0.2], seed=100)
    a, chia dữ liệu học và dữ liệu kiểm tra
    b, chạy chương trình học
    c, tạo dữ liệu ngẫu nhiên cho dữ liệu học và dữ liệu kiểm tra

d, chạy chương trình dự đoán

12. label và feature của câu lệnh bên dưới có nghĩa là gì
LogisticRegression(labelCol = “label” , featuresCol = “features”, maxIter = 10)
a, dữ liệu đầu vào được gán là feature và dự đoán được gán vào label
b, dữ liệu đầu vào được gán là label và kết quả của dữ liệu đầu vào đó được gán vào feature
c, dữ liệu đầu vào được gán là feature và kết quả của dữ liệu đầu vào được gán vào label
d, dữ liệu đầu vào được gán là lebl và kết quả dự đoán được gán vào feature

13. Đâu là lệnh lưu trữ dữ liệu ra ngoài chương trình Spark:
a, input.saveAsTextFile(‘file:///usr/momoinu/mon_loz/hihi.txt’)
b, input.saveAsTextFile(‘/usr/momoinu/mon_loz/hihi.txt’)
c, input.saveAs (‘file:///usr/momoinu/mon_loz/hihi.txt’)
d, input.saveAsTextFile: ‘file:///usr/momoinu/mon_loz/hihi.txt’

14. Đâu là cách submit đúng 1 job lên Spark cluster hoặc chế độ local
a, ./spark-submit wordcount.py README.md
b, ./spark-submit README.md wordcount.py
c, spark-submit README.md wordcount.py
d, phương án a và c

15. Câu lệnh MapReduce trong Spark dưới đây, chia mỗi dòng thành từ dựa vào delimiter nào

```
input.flatMap( lambda x: x.split(“\t”) ).map(lambda x: (x, 1)).reduceByKey(add)
```

a, Tab
b, Dấu cách
c, Dấu hai chấm
d, Dấu phẩy

16. Cơ chế chịu lỗi của datanode trong HDFS
a, dử dụng ZooKeeper để quản lý các thành viên datanode trong cụm
b, sử dụng cơ chế heartbeat, định kỳ các datanode thông báo về trạng thái cho Namenode
c, sử dụng cơ chế heartbeat, Namenode định kỳ hỏi các datanode về trạng thái tồn tại của datanode

17. Cơ chế tổ chức dữ liệu của Datanode trong HDFS
    a, các chunk là các tệp tin trong hệ thống tệp tin cục bộ của máy chủ datanode
    b, các chunk là các vùng dữ liệu liên tục trên ổ cứng của máy chủ data node
    c, các chunk được lưu trữ tin cậy trên datanode theo cơ chế RAID

18. Cơ chế nhân bản dữ liệu trong HDFS
    a, Namenode quyết định vị trí các nhân bản của các chunk trên các datanode
    b, Datanode là primary quyết định vị trí các nhân bản của cac chunk tại các secondary datanode
    c, Client quyết định vị trí lưu trữ các nhân bản với từng chunk

19. HDFS giải quyết bài toán single-point-of-failure cho Namenode bằng cách nào
    a, sử dụng thêm secondary namenode theo cơ chế active-active. Cả Namenode và Secondary Namenode cùng online trong hệ thống
    b, Sử dụng Secondary namenode theo cơ chế active-passive. Secondary namenode chỉ hoạt động khi có vấn đề với namenode
    c… (mất hình)

20. Các mục tiêu chính của Apache Hadoop
    a, lưu trữ dữ liệu khả mở
    b, xử lý dữ liệu lớn mạnh mẽ
    c, trực quan hóa dữ liệu hiệu quả
    d, lưu trữ dữ liệu khả mở và xử lý dữ liệu lớn mạnh mẽ
    e, lưu trữ dữ liệu khả mở, xử lý dữ liệu lớn mạnh mẽ và trực quan hóa dữ liệu hiệu quả

21. phát biểu nào **không đúng** vè Apache Hadoop
    a, xử lý dữ liệu phân tán với mô hình lập trình đơn giản, than thiện hơn như MapReduce
    b, Hadoop thiết kế để mở rộng thông qua kỹ thuật scale-outm tăng số lượng máy chủ
    c, thiết kế để vận hàn trên phần cứng phổ thông, có khả năng chống chịu lỗi phần cứng
    d, thiết kế đẻ vận hành trên siêu máy tính, cấu hình mạnh, độ tin cậy cao

22. Thành phần nào không thuộc thành phần lõi của Hadoop
    a, Hệ thống tập tin phân tán HDFS
    b, MapReduce Framework
    c, YARN: yet another resource negotiator
    d, Apache ZooKeeper
    e, Apache Hbase

23. Spark có thể chạy ở chế độ nào khi chạy trên nhiều máy
    a, chạy trên YARN
    b, chạy trên ZooKeeper
    c, phương án a và b đều sai
    d, phương án a và b đều đúng

24. phát biểu nào sau đây sai về kafka
    a, partition được nhân bản ra nhiều brokers
    b, message sau khi được tiêu thụ (consume) thì không bị xóa.
    c, các topic gồm nhiều partitions
    d, Kafka đảm bảo thứ tự của các message với mỗi topics

25. Mô tả cach thức một client đọc dữ liệu trên HDFS
    a, client thông báo tới namenode để bắt đầu quá trình đọc sau đó client chạy truy vấn các datanode để trực tiếp đọc các chunks
    b, client truy vấn Namenodê để biết được vị trí các chunks. Nếu namenode không biết thì namenode sẽ hỏi các datanode., Sau đó namenode gửi lại thông tin vị trí các chunk cho client. client kết nối song song tới các Datanode để đọc các chunk
    c, client truy vấn namenode để đưa thông tin về thao tác đọc, Namenode kết nối song song tới các datanode để lấy dữ liệu, sau đó trả về cho client
    d, client truy vấn namenode để biết được vị trí các chunks. Namenode trả về vị trí các chunks. Client kết nối song song tới các datanode để đọc các chunks

26. Đâu không phải là tính năng mà NoSQL nào cũng đáp ứng
    a, tính sẵn sàng cao
    b, khả năng mở rộng linh hoạt
    c, phù hợp với dữ liệu lớn

27. Hệ thống nào cho phép đọc ghi dữ liệu tại vị trí ngẫu nhiên, thời gian thực tới hàng terabyte dữ liệu
a, Hbase
b, Flume
c, Pig
d, HDFS

28. Phát biểu nào đúng về Quorum trong Amazon DynamoDB
a, với N là tổng số nhân bản, R là số nhân bản cần đọc trong 1 thao tác đọc. W là số nhân bản cần ghi trong 1 thao tác ghi. N > R + W
b, với N là tổng số nhân bản, R là số nhân bản cần đọc trong 1 thao tác đọc. W là số nhân bản cần ghi trong 1 thao tác ghi. N < R + W
c, với N là tổng số nhân bản, R là số nhân bản cần đọc trong 1 thao tác đọc. W là số nhân bản cần ghi trong 1 thao tác ghi. N = R + W

29. Phát biểu nào sau đây sai về Kafka
a, nhiều consumer có thể cùng đọc 1 topic
b, 1 message có thể được đọc bởi nhiều consumer khác nhau
c, số lượng consumer phải ít hơn hoặc bằng số lượng partitions
d, 1 message chỉ có thể được đọc bởi 1 consumer trong 1 consumer group

30. Đâu là một dạng của NoSQL
a, MySQL
b, JSON
c, Key-value store
d, OLAP.

31. phát biểu nào sai về Presto
a, presto là một engine truy vấn SQL hiệu năng cao, phân tán cho dữ liệu lớn.
b, presto thích hợp với các công cụ Business Intelligence
c, presto được quản lý bởi presto software foundation
d, presto được quản lý bởi apache software foundation

32. Phát biểu nào đúng về Presto

a, các stage được thực thi theo cơ chế pipeline, không có thời gian chờ giữa các stage như Map Reduce

b, Presto cho phép xử lý kết tập dữ liệu mà kích thước lớn hơn kích thước bộ nhớ trong

c, Presto có cơ chế chịu lỗi khi thực thi truy vấn

33. Chọn phát biểu đúng khi nói về MongoDB

a, MongoDB có các trình điều khiển driver cho nhiều ngôn ngữ lập trình khác nhau.

b, các văn bản có thể chứa nhiều cặp key-value hoặc key-array, hoặc các văn bản lồng (nested documents)

c, tất cả các phương án trên

d, MongoDB hay các NoSQL có khả năng khả mở tốt hơn các CSDL quan hệ truyền thống

34. Giữa Pig và Hive, công cụ nào có giao diện truy vấn gắn với ANSI SQL hơn

a. Pig

b. không phải 2 đáp án trên

c. Hive

35. Công ty nào đã phát triển Apache Cassandra giai đoạn đầu tiên

a, Google

b, twitter

c, linkedin

d, facebook

36. Phát biểu về định lý CAP

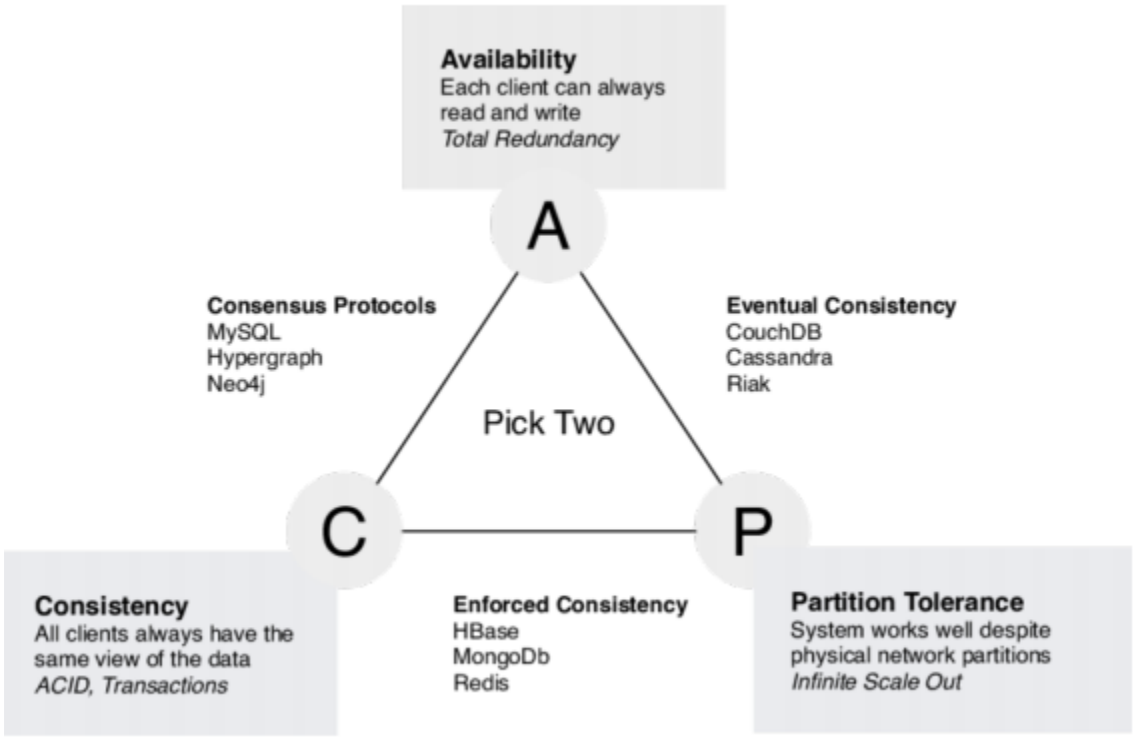
The limitations of distributed databases can be described in the so called the CAP theorem

• Consistency: every node always sees the same data at any given instance (i.e., strict consistency)

• Availability: the system continues to operate, even if nodes in a cluster crash, or some hardware or software parts are down due to upgrades

• Partition Tolerance: the system continues to operate in the presence of network partitions

CAP theorem: any distributed database with shared data, can have *at most two* of the three desirable properties, C, A or P. These are trade-offs involved in distributed system by Eric Brewer in PODC 2000.



37. Giải thích ngắn gọn về các phương pháp tổng quát trong việc thích nghi các giải thuật học máy cho dữ liệu lớn

38. Cho đoạn chương trình sau:

rdd = sc.parallelize([“hello”, “world”, “good”, “hello”], 2)
rdd = rdd.map(lambda w : (w, 1))
Đưa ra kết quả của rdd.glom().collect()

```
rdd = sc.parallelize(["hello", "world", "good", "hello"], 2)
rdd = rdd.map(lambda w : (w, 1))
result=rdd.glom().collect()
print(result)
```

```
[[('hello', 1), ('world', 1)], [('good', 1), ('hello', 1)]]
```

```
import numpy as np
import os

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
sc = SparkContext('local')
spark = SparkSession(sc)
sqlContext = SQLContext(sc)
```

```
text_file=sc.textFile("a/*.txt")
counts = text_file.flatMap(lambda line: line.split("\t")) \
             .map(lambda word: (word, 1)) \
             .reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("e")
```

```
rdd = sc.parallelize(["hello", "world", "good", "hello"], 2)
rdd = rdd.map(lambda w : (w, 1))
result=rdd.glom().collect()
print(result)
```

```
[[('hello', 1), ('world', 1)], [('good', 1), ('hello', 1)]]
```

import numpy as np
import os

```
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
sc = SparkContext('local')
spark = SparkSession(sc)
sqlContext = SQLContext(sc)

rdd = sc.parallelize(["hello", "world", "good", "hello"], 2)
rdd = rdd.map(lambda w : (w, 1))
result=rdd.glom().collect()
print(result)
```

39. Giả sử một textfile kích thước lớn được đặt ở đường dẫn hdfs://user/bigfile.txt. Viết chương trình Spark đếm số lần xuất hiện của chuỗi “big data processing” trong file text này.

```
linesRDD = sc.textFile("hdfs://user/bigfile.txt")
```

```
df=spark.read.format("json").load("../data/flight-data/json/2015-summary.json")
```

```
df = (spark.read.format("com.databricks.spark.csv")
      .option("header", "true")
      #.option("inferSchema","true")
      .schema(schema)
```

```
    .load("hdfs://192.168.56.10:9000/user/output/data10/*.csv"))
df.printSchema()
df.createOrReplaceTempView("dfTable")
print("Số  bản ghi:")
print(df.count())
df.show(20, False)
```

40. Cho đoạn chương trình sau

```
def mystr(d):
    return d

def myconcat(a, b):
    return a + b

def mypartConcat(a, b):
    return a + b

rdd = sc.parallelize([  (“a”, 1), (“b”, 1), (“a”, 2), (“a”, 8), (“c”, 4), (“a”, 12),
(“a”, 18), (“c”, 14)  ], 3)
```

Tinhs kết quả của rdd.combineByKey(mystr, myconcat,

(đcm câu này thiếu dòng cuối, gõ đến đây ms thấy)