

CƠ CHẾ MÀ NOSQL SỬ DỤNG ĐỂ TĂNG KHẢ NĂNG CHỊU LỖI

☒ a. REPLICATION

☐ b. GIAO DIỆN TRUY VẤN ĐƠN GIẢN HƠN SO VỚI CSQL QUAN HỆ TRUYỀN THỐNG

☐ c. PHÂN MẢNH VÀ PHÂN TÁN DỮ LIỆU RA NHIỀU MÁY CHỦ

☐ d. DÙNG RAID

us page

NoSQL databases (Not only SQL) are a class of non-relational databases that provide a flexible and scalable way to store and manage data. NoSQL databases use various mechanisms to improve fault tolerance, including:

Data partitioning: NoSQL databases can partition data across multiple nodes in a cluster, allowing the database to scale horizontally. By distributing data across multiple nodes, NoSQL databases can handle large amounts of data and provide high availability in the event of node failures.

Replication: NoSQL databases can replicate data across multiple nodes in a cluster, improving fault tolerance and ensuring data availability in the event of node failures. Replication can be performed synchronously or asynchronously, and data can be replicated across multiple data centers to improve disaster recovery.

ữ và xử lý dữ liệu lớn

Question 13

Not yet
answered

Marked out of
1.00

ĐÂY LÀ MỘT CSDL DẠNG CỘT MỞ RỘNG?

- ☒ a. CASSANDRA
☐ b. REDIS
☐ c. ELASTICSEARCH
☐ d. MONGODB

Previous page

u trữ và xử lý dữ liệu lớn

Time left 0:19:35

Question 11

Not yet
answered
Marked out of
1.00

ĐÂY LÀ CHƯƠNG TRÌNH WORDCOUNT DÙNG, VỚI INPUT LÀ MỘT RDD CHỨA DỮ LIỆU ĐỌC TỪ FILE VĂN BẢN

- ☐ a. `INPUT.FLATMAP(LAMBDA X: X.SPLIT(" ")).MAP(LAMBDA X: (X, 1)).REDUCEBYVALUE(LAMBDA X, Y: X+Y).SORTBY(LAMBDA X: X[1])`
- ☐ b. `INPUT.MAP(LAMBDA X: X.SPLIT(" ")).FLATMAP(LAMBDA X: (X, 1)).REDUCEBYKEY(LAMBDA X, Y: X+Y).SORTBY(LAMBDA X: X[1])`
- ☐ c. `INPUT.MAP(LAMBDA X: (X, 1)).REDUCEBYKEY(LAMBDA X, Y: X+Y).SORTBY(LAMBDA X: X[1])`
- ☒ d. `INPUT.FLATMAP(LAMBDA X: X.SPLIT(" ")).MAP(LAMBDA X: (X, 1)).REDUCEBYKEY(LAMBDA X, Y: X+Y).SORTBY(LAMBDA X: X[1])`

Previous page

```
words = input_file.flatMap(lambda line: line.split(" ")) # Map
each word to a tuple of (word, 1) and then reduce by key to
count the occurrences of each word
word_counts = words.map
(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b)
```

và xử lý dữ liệu lớn

Question 7

Answer saved
Marked out of
1.00

CSDL NÀO PHÙ HỢP VỚI DỮ LIỆU MẠNG XÃ HỘI, DỮ LIỆU CÓ SỰ LIÊN KẾT

- ☒ a. DOCUMENT STORE
- ☐ b. KEY-VALUE
- ☐ c. COLUMNAR STORE
- ☒ d. GRAPH STORE

Clear my choice

Previous page

Mudim v0.8 • Tắt • VNI • Telex • Viqr • Tổng hợp • Tự động • Chính tả • Bỏ dấu kiểu mới [Bật/Tắt (F9)

xử lý dữ liệu lớn

Question **23**

Answer saved

Marked out of
1.00

SPARK HỖ TRỢ CÁC CLUSTER MANAGER NÀO?

- ☐ a. MESOS
- ☒ b. TẤT CẢ CÁC ĐÁP ÁN TRÊN
- ☐ c. STANDALONE CLUSTER MANAGER
- ☐ d. YARN

[Clear my choice](#)

[Previous page](#)

1 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 hoá 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 hoá dữ liệu hiệu quả~~

2 Phát biểu nào sau đây không đúng về Apache Hadoop

- A. Xử lý dữ liệu phân tán với mô hình lập trình đơn giản, thân thiện hơn như MapReduce
- B. Hadoop thiết kế để mở rộng thông qua kỹ thuật scale-out, tăng số lượng máy chủ
- C. Thiết kế để vận hành trên phần cứng phổ thông, có khả năng chống chịu lỗi phần cứng
- D. Thiết kế để vận hành trên ~~siêu~~ máy tính, cấu hình mạnh, độ tin cậy cao ✓

3 Thành phần nào không thuộc thành phần lõi của Hadoop

- A. Hệ thống tệp tin phân tán HDFS
- B. Mapreduce framework |
- C. YARN: yet another resource negotiator
- D. Apache Zookeeper ✓
- E. Apache Hbase ✓

4 Hadoop giải quyết bài toán khả mở bằng cách nào? Chọn đáp án sai.

- A. Thiết kế hướng 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

5 Hadoop giải quyết bài toán chịu lỗi thông qua kỹ thuật gì. Chọn đáp án sai

- A. Hadoop chịu lỗi thông qua 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.

6 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

7 Mô tả cách thức một client đọc dữ liệu trên HDFS

- A. 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 chunk ✓
- B. Client thông báo tới namenode để bắt đầu quá trình đọc sau đó client truy vấn các datanode để trực tiếp đọc các chunks
- 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. Nếu Namenode không biết về vị trí các chunk 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.

8 Mô tả cách thức một client ghi dữ liệu trên HDFS

- A. Client kết nối tới Namenode chỉ định muốn ghi vào chunk nào. Namenode trả về vị trí các chunk cho client. Client ghi đồng thời vào các datanode.
- B. Client kết nối tới Namenode chỉ định khối lượng dữ liệu cần ghi. Namenode trả về vị trí các chunk cho client. Client ghi chunk tới datanode đầu tiên, sau đó các datanode tự động thực thi nhân bản. Quá trình ghi kết thúc thì tất cả các chunk và các nhân bản đã được ghi thành công. ✓
- C. Client kết nối tới Namenode chỉ định khối lượng dữ liệu cần ghi. Namenode trả về vị trí các chunk cho client. Client ghi đồng thời các chunk vào datanode. Với mỗi chunk, các datanode thực thi nhân bản tự động sau khi thao tác ghi thành công.

9 Cơ chế chịu lỗi của datanode trong HDFS

- A. Sử 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 các datanode.

10 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 datanode.
- C. Các chunk được lưu trữ tin cậy trên datanode theo cơ chế RAID.

11 Cơ chế nhân bản dữ liệu trong HDFS

- A. Namenode quyết định vị trí các nhân bản của các chunk trên datanode. ✓
- B. Datanode là primary quyết định vị trí các nhân bản của các chunk tại các secondary datanode.
- C. Client quyết định vị trí lưu trữ các nhân bản với từng chunk. I

12 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. ✓

Câu 1: Đầu vào dữ liệu cho chương trình Spark có thể là:

- A. Local file
- B. HDFS, NFS
- C. Amazon S3, Elasticsearch
- D. Cả 3 phương án trên ✓

Câu 2: Đầu là lệnh lưu dữ liệu ra ngoài chương trình Spark:

- A. input.saveAsTextFile('file:///usr/zeppelin/notebook/dataset/new.txt') ✓
- B. input.saveAsTextFile('/usr/zeppelin/notebook/dataset/new.txt')
- C. input.saveAs('file:///usr/zeppelin/notebook/dataset/new.txt')
- D. input.saveAsTextFile('file:///usr/zeppelin/notebook/dataset/new.txt')

Câu 3: Đầu là cách submit đúng một 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

Câu 4: 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

THUẬT TOÁN VÀ LƯỚI
Cầu 1: Đầu vào dữ liệu cho chương trình Spark có thể là:

- A. Local file
- B. HDFS, NFS
- C. Amazon S3, Elasticsearch
- D. Cả 3 phương án trên

Cầu 2: Đầu là lệnh lưu dữ liệu ra ngoài chương trình Spark:

- A. `input.saveAsTextFile("file:///usr/zeppelin/notebook/dataset/new.txt")`
- B. `input.saveAsTextFile("/usr/zeppelin/notebook/dataset/new.txt")`
- C. `input.saveAsTextFile("file:///usr/zeppelin/notebook/dataset/new.txt")`
- D. `input.saveAsTextFile("file:///usr/zeppelin/notebook/dataset/new.txt")`

Cầu 3: Đầu là cách submit đúng một 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

Cầu 4: Câu lệnh MapReduce trong Spark dưới đây, chia mỗi dòng thành từ dựa vào delimiter nào.

`Input.flatMap(lambda x: x.split("\r")).map(lambda x: [x, 1]).reduceByKey(add)`

- A. Tab
- B. Dấu cách
- C. Dấu hai chấm
- D. Dấu phẩy

Module 12&13

Câu 5: Data Pipeline nào sau đây là đúng trên Spark

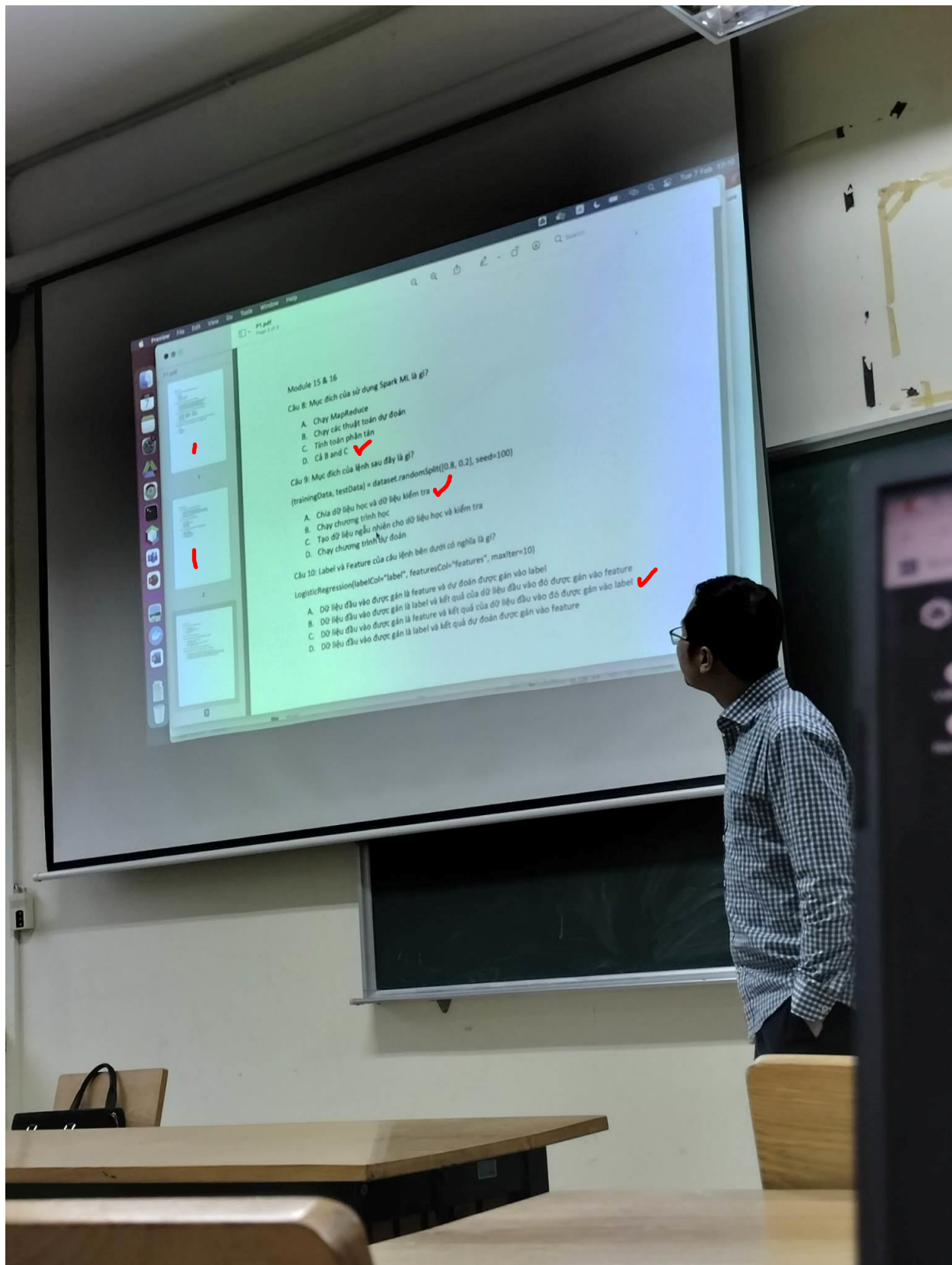
- A. Spark → RabbitMQ → Elasticsearch → Hiển thị ✓
- B. Dữ liệu sensor → RabbitMQ → Elasticsearch → Spark → Hiển thị
- C. Dữ liệu sensor → Elasticsearch → RabbitMQ → Spark → Hiển thị
- D. Spark → Elasticsearch → Hiển thị ✓

Câu 6: Mục đích của sử dụng RabbitMQ là gì?

- A. Lưu trữ dữ liệu
- B. Tránh dữ liệu bị mất mát ✓
- C. Hiển thị dữ liệu
- D. Phân tích dữ liệu

Câu 7: 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. Cả 2 phương án A và B ✓



1. đâu là kĩ thuật có thể dùng để thích nghi các giải thuật học máy cho dữ liệu lớn:

A: sub-sampling, principal componenet analysis, feature extraction và feature selection

B song song hóa trên mapreduce hay spark

C các kiến trúc mới xử lý luồng liên tục như mini-batch, complex event processing

D tất cả

E ý B, C

2. Các mục tiêu chính của apache hadoop

A lưu trữ dữ liệu khá mở

B Xử lý dữ liệu lớn mạnh mẽ

C trực quan hóa dữ liệu hiệu quả

D lưu trữ dữ liệu khá mở và xử lý dữ liệu lớn mạnh mẽ

E lưu trữ dữ liệu khá mở và xử lý dữ liệu lớn mạnh mẽ và trực quan hóa dữ liệu hiệu quả

3. Phạt biểu sai về hadoop

A. xử lý dữ liệu phân tán với mô hình lập trình đơn giản, thân thiện hơn như mapreduce

B, Hadoop thiết kế để mở rộng thông qua kỹ thuật scale-out, tăng số lượng máy chủ

C. Thiết kế để vận hành trên phần cứng phổ thông, có khả năng chống chịu lỗi phần cứng

D. Thiết kế để vận hành trên siêu máy tính, cấu hình mạnh, độ tin cậy cao

Đáp án: D

Đề: cơ chế chịu lỗi của datanode trong HDFS?

A sử dụng zookeeper để quản lý các thành viên datanode trong cụm

B Sử dụng cơ chế heartbeat, định kì các datanode thông báo về trạng thái cho Namenode

C Sử dụng cơ chế heartbeat, Namenode định kì hỏi các datanode về trạng thái tồn tại của các datanode

Đề: 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 datanode

C các chunk được lưu trữ tin cậy trên datanode theo cơ chế RAID

Đề: Cơ chế nhân bản dữ liệu trong HDFS

A. Namenode quyết định vị trí các nhân bản của các chunk trên datanode



B. Datanode là primary quyết định vị trí các nhân bản của các chunk tại các secondary datanode

C. Client quyết định vị trí lưu trữ các nhân bản với từng chunk