Đề kiểm tra giữa kỳ

Học phần Quản trị Dữ liệu lớn

(Thời gian làm bài: 120 phút)

(Sinh viên không được sử dụng bất kì tài liệu nào, trao đổi bài hay truyền tải hình ảnh / nội dung liên quan đến bài thi thì ngay lập tức hủy kết quả bài thi)

Câu 1: (2 điểm)

(1) Login vào hệ thống : ssh your_student_id@103.143.206.51 -p 21122

(2) Tạo thư mục mid_term trong thư mục home trên hệ thống sandbox của mình

(3) Trong thư mục mid_term tạo 3 thư mục có tên lần lượt là a, b, c

(4) Tạo file có tên a.txt trong thư mục a với nội dung trong file là nội dung thư mục /usr

(5) Copy file u.data, u.item, u.genre vào thư mục b

(6) Trên hệ thống HDFS tạo thư mục mid_term. Trong thư mục tạo thêm 3 thư mục lần lượt là a, b, c

(7) Copy file a.txt từ hệ thống sandbox lên thư mục a trên hệ thống HDFS

(8) Copy file u.data, u.item, u.genre từ hệ thống sandbox vào thư mục b trên hệ thống HDFS

Câu 2: (2 điểm)

Trong thư mục b:

(1) Viết chương trình java thực hiện MapReduce trên Hadoop để tính điểm đánh giá trung bình cho từng phim. Yêu cầu chỉ tính những phim có điểm đánh giá là "số nguyên tố"

(2) Dịch chương trình trên hệ thống Sandbox-hdp

(3) Tạo biến môi trường và đóng gói thành một file .jar

(4) Chạy MapReduce Job và kiểm tra kết quả

Câu 3: (2 điểm)

Trong thư mục c:

(1) Tạo một Pig script có tên là “abc.pig”

(2) Tính tổng số điểm đánh giá phim của từng user đánh giá cho các phim

(Ví dụ: user50 rate movie10 : 3*

user50 rate movie15 : 2*

=> Tổng số lượt rate của user50 là 5)

(3) Lọc tất cả user có tổng điểm đánh giá > 250. Sao cho kết quả trả về có dạng {userID: int, sum: long}

(4) Lưu bảng kết quả vào ‘/user/your_id/mid_term/c’

(5) Copy file "part-r-00000" về thư mục ./mid_term/c trên hệ thống Sandbox-hdp

Câu 4: (4 điểm)

Sử dụng PySpark trên Google Colab thực hiện các yêu cầu sau:

(1) Tạo một Spark RDD chứa thông tin movieID, userID, và rating từ file u.data

(2) Tạo một list các dictionary: {moviID: “movie name") từ file u.item

(3) Tạo một Spark RDD, X1, chứa thông tin movieID và one-hot vector chứa genre của movie từ file u.item

(4) Viết hàm Python để tính Cosine Similarity của 2 movies bất kỳ từ X1

(5) Tạo một Spark RDD, X2, chứa thông tin movieID và hot-vector chứa điểm rate của các users

(6) Viết hàm python tính Cosine Similarity của 2 movies bất kỳ từ X2. So sánh với phương pháp tính bên trên