

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
VIỆN TRÍ TUỆ NHÂN TẠO**

-----***-----

**BÁO CÁO MÔN HỌC KỸ THUẬT VÀ CÔNG NGHỆ DỮ LIỆU LỚN
ĐỀ TÀI
THUẬT TOÁN KMEAN TRONG PHÂN CỤM ẢNH**

Nhóm sinh viên thực hiện:

1. Nguyễn Việt Vũ-22022632
2. Nguyễn Tông Quân-22022635
3. Nguyễn Xuân Trình-22022558
4. Phạm Đăng Phong-22022614

Giảng viên hướng dẫn: TS. Trần Hồng Việt

GV. Lương Sơn Bá

HÀ NỘI, 12/2024

MỞ ĐẦU

Công nghệ Big Data đã tạo ra một xu hướng mạnh mẽ, hỗ trợ trích xuất thông tin quý giá trong nhiều lĩnh vực như y tế, giao thông, và xử lý ảnh. Đi kèm với đó, các framework như **Hadoop** cùng mô hình **MapReduce** ngày càng được ứng dụng rộng rãi để xử lý dữ liệu lớn.

Trong bài toán phân cụm dữ liệu, thuật toán **K-Means** đã chứng minh hiệu quả trong việc nhóm các đối tượng có đặc trưng tương đồng. Khi kết hợp với MapReduce, K-Means có thể mở rộng để xử lý các tập dữ liệu ảnh lớn, mang lại giá trị thực tiễn cao.

Từ đó, chúng em đã chọn đề tài: "**Thuật toán K-Means & lập trình MapReduce hóa trong phân cụm ảnh**" để làm báo cáo môn học.

Báo cáo gồm 5 chương:

- **Chương 1:** Tổng quan về dữ liệu lớn.
- **Chương 2:** Phân cụm dữ liệu bằng thuật toán K-Means.
- **Chương 3:** MapReduce thuật toán K-Means trong phân cụm ảnh.
- **Chương 4:** Cải tiến thuật toán
- **Chương 5:** Kết luận và hướng phát triển.

MỤC LỤC

VIỆN TRÍ TUỆ NHÂN TẠO	1
BÁO CÁO MÔN HỌC KỸ THUẬT VÀ CÔNG NGHỆ DỮ LIỆU LỚN	1
MỞ ĐẦU	2
CHƯƠNG 1: TỔNG QUAN VỀ DỮ LIỆU LỚN	4
1.1 Định nghĩa.	5
1.2 Đặc trưng cơ bản của dữ liệu lớn.	5
1.3 Tổng quan về Hadoop.	6
1.4 Tổng quan về Hadoop.	7
CHƯƠNG 2: PHÂN CỤM DỮ LIỆU BẰNG THUẬT TOÁN KMEANS	9
2.1 Giới thiệu thuật toán Kmeans.	9
2.2 Triển khai thuật toán phân cụm Kmeans.	9
2.3 Ví dụ minh họa thuật toán.	11
CHƯƠNG 3: ỨNG DỤNG MAPREDUCE KMEANS TRONG PHÂN CỤM HÌNH ẢNH	12
3.1 Ý tưởng MapReduce Kmeans trong phân cụm ảnh	12
3.2 Lưu đồ của thuật toán MapReduce Kmeans	12
3.3 Giải pháp MapReduce Kmeans trong phân cụm ảnh	14
3.4 Demo chương trình cài đặt.	16
3.4.1 Demo cài đặt hadoop thành công.	16
3.4.2 Demo Chương trình demo.	17
CHƯƠNG 4: CẢI TIẾN THUẬT TOÁN	21
4.1 Bổ sung hàm khoảng cách:	21
4.2 Cải tiến thuật toán phân cụm Kmeans++:	21
4.3 Thử nghiệm với HOG:	22
4.4 Kết quả và đánh giá:	22
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	24
5.1 Ứng dụng thực tế của thuật toán:	24
Phân tích và tối ưu hóa ảnh trong ứng dụng trực tuyến	24
Phân tích ảnh vệ tinh và địa lý	24
5.2 Kết luận	25
5.3 Hướng phát triển	25
TÀI LIỆU THAM KHẢO	26
NHIỆM VỤ CỦA CÁC THÀNH VIÊN	27

CHƯƠNG 1: TỔNG QUAN VỀ DỮ LIỆU LỚN

1.1 Định nghĩa.

Theo wikipedia: Dữ liệu lớn là thuật ngữ dùng để mô tả các bộ dữ liệu có kích thước hoặc độ phức tạp vượt quá khả năng xử lý của các phương pháp truyền thống.

Theo **Gartner**, dữ liệu lớn bao gồm các nguồn thông tin với ba đặc điểm chính:

khối lượng lớn, tốc độ xử lý cao, và định dạng đa dạng. Để khai thác hiệu quả, cần áp dụng các phương pháp mới nhằm hỗ trợ việc ra quyết định, khám phá và tối ưu hóa quy trình.

Dữ liệu lớn có thể đến từ nhiều nguồn khác nhau, chẳng hạn như giao dịch trực tuyến, mạng xã hội, cảm biến IoT, hoặc các tài liệu, hình ảnh được chia sẻ trên internet.

 *Big Data Source Analysis*



Hình 1. Minh họa nguồn gốc của dữ liệu.

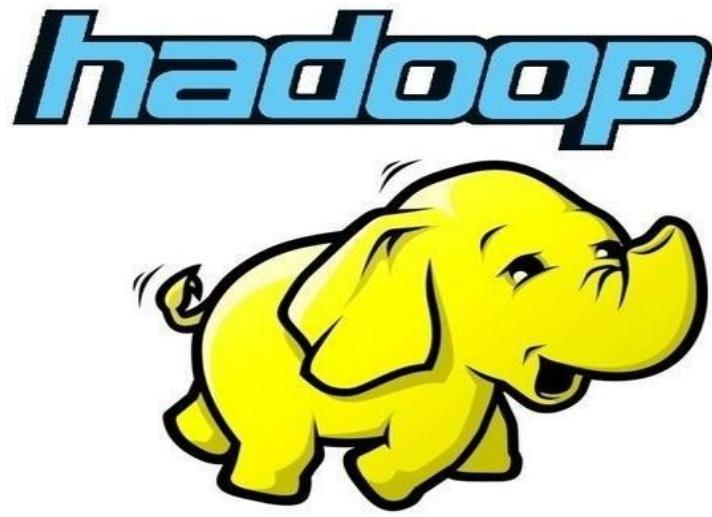
Một số lợi ích nổi bật mà dữ liệu lớn mang lại bao gồm: **cắt giảm chi phí, tiết kiệm thời gian, tối ưu hóa sản phẩm**, và hỗ trợ con người trong việc đưa ra các quyết định chính xác, hợp lý hơn.

1.2 Đặc trưng cơ bản của dữ liệu lớn.

- (1) *Khối lượng lớn (Volume)*: Lượng dữ liệu rất lớn và không ngừng tăng lên. Đến năm 2014, khối lượng này đã có thể đạt đến hàng trăm terabyte.
- (2) *Tốc độ (Velocity)*: Dữ liệu được tạo ra và truyền tải với tốc độ cao, đòi hỏi hệ thống phải xử lý gần như theo thời gian thực.
- (3) *Đa dạng (Variety)*: Hơn 80% dữ liệu ngày nay là phi cấu trúc, bao gồm tài liệu, blog, hình ảnh, video, và các dạng thông tin khác.
- (4) *Độ tin cậy/chính xác (Veracity)*: Việc loại bỏ dữ liệu không chính xác, nhiễu loạn, và đảm bảo tính chính xác trong phân tích là một thách thức quan trọng của dữ liệu lớn.
- (5) *Giá trị (Value)*: Giá trị mà dữ liệu mang lại phụ thuộc vào khả năng khai thác thông tin và ứng dụng chúng vào thực tế, góp phần gia tăng hiệu quả kinh doanh hoặc cải thiện các quyết định chiến lược.

1.3 Tổng quan về Hadoop.

Apache Hadoop là một nền tảng phần mềm mã nguồn mở, được thiết kế để hỗ trợ phát triển và triển khai các ứng dụng xử lý dữ liệu phân tán trên các cụm máy tính sử dụng phần cứng phổ thông.

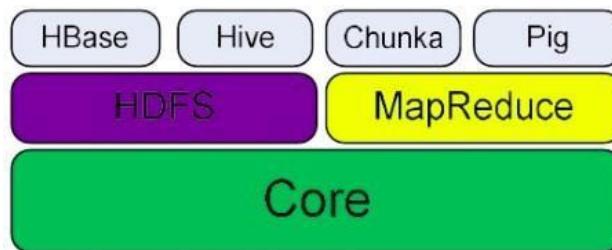


Hình 2: Logo của Hadoop

Hadoop bao gồm nhiều thành phần quan trọng, trong đó đáng chú ý nhất là:

- **HDFS (Hadoop Distributed File System):** Hệ thống file phân tán cho phép lưu trữ dữ liệu trên nhiều node khác nhau, đảm bảo tính sẵn sàng và khả năng chịu lỗi cao.
- **MapReduce:** Mô hình lập trình hỗ trợ xử lý dữ liệu bằng cách chia nhỏ công việc thành các phần và chạy song song trên các node trong cụm.

Ngoài ra, Hadoop còn tích hợp các công cụ khác như HBase, Hive, Pig, và Chunka, giúp mở rộng khả năng xử lý dữ liệu theo nhiều cách khác nhau.



Hình 3. Thành phần của Hadoop

Hadoop được thiết kế để hiện thực hóa mô hình Map/Reduce, trong đó:

- Dữ liệu được phân chia thành các phân đoạn nhỏ và xử lý song song trên các node.
- HDFS hỗ trợ lưu trữ dữ liệu với cơ chế phân tán, đồng thời tự động quản lý và

khắc phục lỗi phần cứng hoặc sự cố trong cụm.

Tóm lại, Hadoop là một framework mạnh mẽ, được viết bằng Java, giúp xử lý hiệu quả các bài toán liên quan đến dữ liệu lớn bằng cách tận dụng sức mạnh của hệ thống phân tán.

1.4 Tổng quan về Hadoop.

Định nghĩa:

Theo Google, **MapReduce** là một mô hình lập trình được thiết kế để xử lý tính toán song song và phân tán trên các hệ thống phân tán.

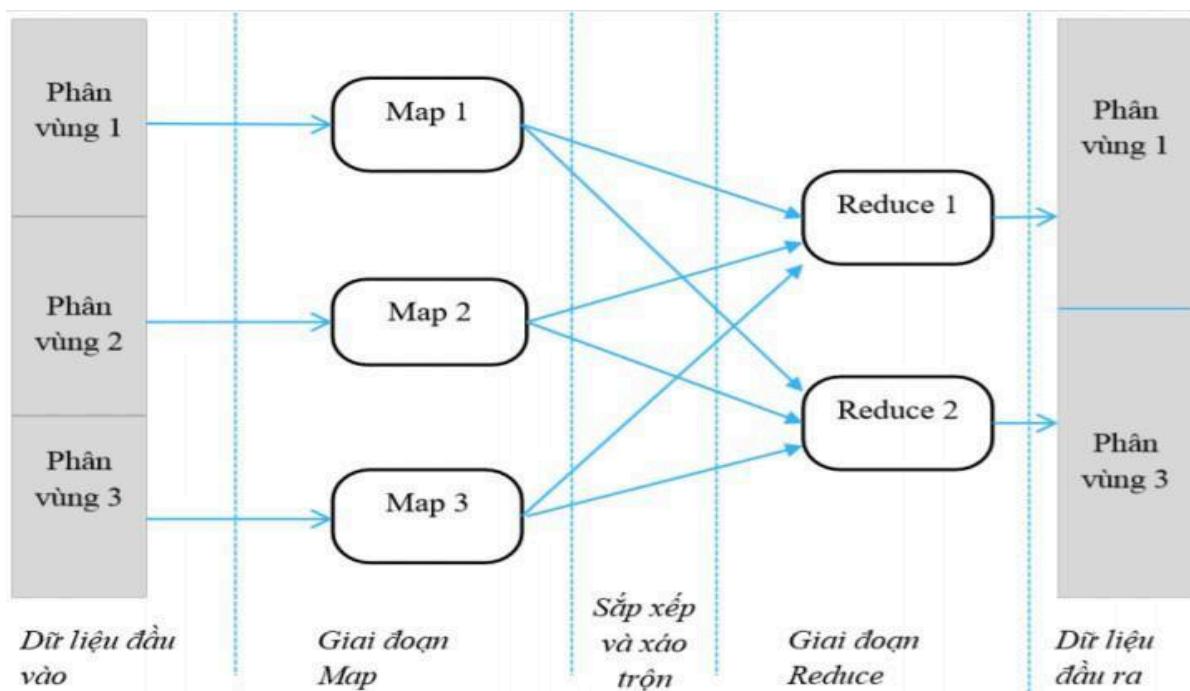
Quy trình xử lý của MapReduce gồm hai bước chính:

1. **Phân rã công việc:** Nhiệm vụ chính được chia thành các công việc nhỏ hơn, sau đó phân phối đến các máy tính trong hệ thống để xử lý song song.
2. **Thu thập kết quả:** Tập hợp kết quả từ các công việc nhỏ, tổng hợp lại để tạo ra kết quả cuối cùng.

Ứng dụng của MapReduce:

- Xử lý các tập dữ liệu có kích thước lớn.
- Thích hợp cho các bài toán cần phân tích, xử lý dữ liệu với thời gian xử lý đáng kể (có thể kéo dài từ vài phút đến vài giờ).

Thực thi mô hình MapReduce:



Hình 4. Thực thi mô hình Mapreduce

Hàm Map:

- Tiếp nhận từng phần dữ liệu đầu vào (input split).
- Thực hiện rút trích thông tin cần thiết từ các phần tử, ví dụ như lọc hoặc trích xuất dữ liệu.
- Kết quả là các cặp giá trị trung gian dạng (*key, value*).

Hàm Reduce:

- Tổng hợp các cặp giá trị trung gian từ hàm Map.
- Thực hiện tính toán và tạo ra kết quả đầu ra cuối cùng.

Mô hình MapReduce đảm bảo rằng công việc được xử lý song song và tối ưu hóa việc sử dụng tài nguyên trong hệ thống phân tán, từ đó cải thiện hiệu suất và khả năng mở rộng.

CHƯƠNG 2: PHÂN CỤM DỮ LIỆU BẰNG THUẬT TOÁN KMEANS

2.1 Giới thiệu thuật toán Kmeans.

Giới thiệu:

Thuật toán **K-Means** được sử dụng phổ biến trong bài toán phân cụm (Clustering). Đây là một thuật toán học không giám sát, có mục tiêu nhóm dữ liệu thành k cụm dựa trên sự tương đồng giữa các điểm dữ liệu. K-Means hoạt động tốt với các thuộc tính dạng số và có khả năng mở rộng với dữ liệu lớn.

Ý tưởng:

Phân cụm dữ liệu bằng cách:

1. Gán mỗi điểm dữ liệu vào một cụm sao cho khoảng cách đến tâm cụm (centroid) là ngắn nhất.
2. Cập nhật lại tâm cụm dựa trên trung bình của các điểm dữ liệu trong cụm.
3. Lặp lại các bước trên cho đến khi các tâm cụm không còn thay đổi đáng kể hoặc đạt ngưỡng hội tụ.

2.2 Triển khai thuật toán phân cụm Kmeans.

Khởi tạo:

- Chọn ngẫu nhiên k điểm dữ liệu từ tập $X = \{x_1, x_2, \dots, x_n\}$ làm tâm cụm ban đầu $C = \{c_1, c_2, \dots, c_k\}$

Phân cụm:

- Gán mỗi điểm dữ liệu x_i vào cụm j dựa trên khoảng cách nhỏ nhất giữa x_i

và tâm cụm c_j :

$$Cluster(x_i) = \arg \min_{j \in \{1, \dots, k\}} d(x_i, c_j)$$

Trong đó:

- $d(x_i, c_j)$ là khoảng cách giữa điểm x_i và tâm cụm c_j (thường sử dụng khoảng cách Euclid):

$$d(x_i, c_j) = \sqrt{\sum_{l=1}^d (x_{il} - c_{jl})^2}$$

Với x_{il} , c_{jl} lần lượt là giá trị của điểm x_i và tâm cụm c_j trên chiều l.

Cập nhật tâm cụm:

- Sau khi gán tất cả các điểm vào cụm, cập nhật tâm cụm c_j bằng trung bình cộng của tất cả các điểm trong cụm đó:

$$c_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$$

Trong đó:

- S_j là tập hợp các điểm thuộc cụm j.
- $|S_j|$ là số lượng điểm trong cụm j.

Hàm mục tiêu (hội tụ):

- Thuật toán hội tụ khi hàm mục tiêu (tổng bình phương khoảng cách từ các điểm đến tâm cụm tương ứng) không thay đổi đáng kể:

$$J(C) = \sum_{j=1}^k \sum_{x_i \in S_j} \|x_i - c_j\|^2$$

Trong đó:

- $\|x_i - c_j\|^2$ là bình phương khoảng cách Euclid.

2.3 Ví dụ minh họa thuật toán.

- Cho tập dữ liệu:
 $X = \{(2, 3), (3, 3), (6, 7), (8, 8)\}, k = 2$
- Khởi tạo ngẫu nhiên $c_1=(2,3), c_2=(6,7)$
- Phân cụm dựa trên khoảng cách Euclid. Ví dụ, khoảng cách giữa $(3,3)$ và $c_1=(2,3)$:

$$d((3, 3), (2, 3)) = \sqrt{(3 - 2)^2 + (3 - 3)^2} = 1$$

- Cập nhật lại tâm cụm:
 Nếu cụm $S_1=\{(2,3),(3,3)\}$, thì:
 $c_1 = \frac{1}{2}((2, 3) + (3, 3)) = (2.5, 3)$

Ưu điểm và hạn chế của K-Means:

- **Ưu điểm:**
 - Đơn giản, dễ triển khai.
 - Thời gian tính toán nhanh với tập dữ liệu lớn.
- **Hạn chế:**
 - Kết quả phụ thuộc vào khởi tạo tâm cụm ban đầu.
 - Khó phân cụm tốt nếu dữ liệu không có ranh giới rõ ràng hoặc phân cụm không đều.

Ứng dụng:

Thuật toán K-Means được áp dụng rộng rãi trong nhiều lĩnh vực, như đồ họa, y tế, nhận diện đối tượng, phân tích khách hàng, và phân tích dữ liệu lớn,...

CHƯƠNG 3: ÚNG DỤNG MAPREDUCE KMEANS TRONG PHÂN CỤM HÌNH ẢNH

3.1 Ý tưởng MapReduce Kmeans trong phân cụm ảnh

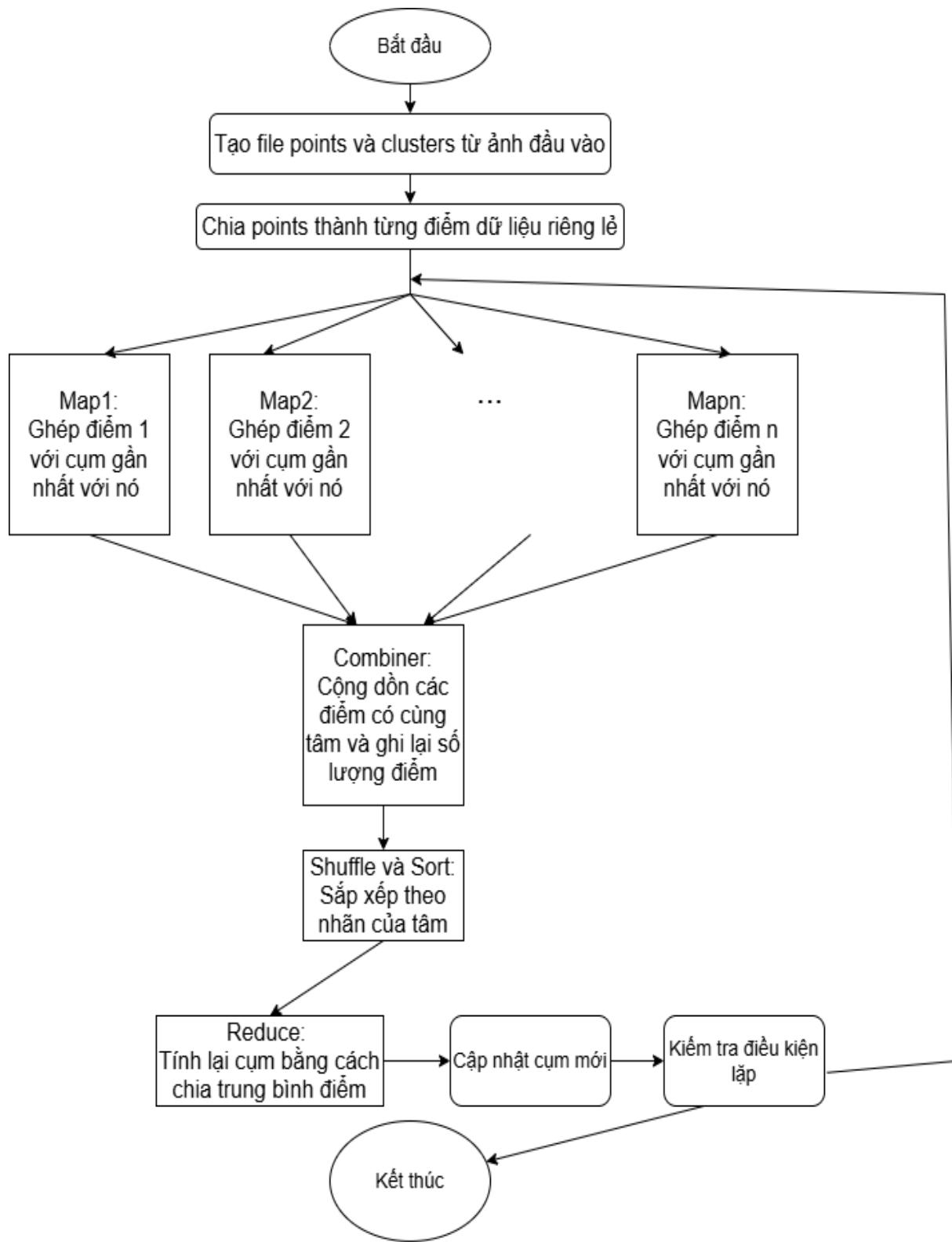
Nhiệm vụ: Phân cụm từng điểm ảnh của một bức ảnh dựa trên dải màu RGB

- Sử dụng thuật toán Kmean clustering để nhóm các pixel có đặc điểm giống nhau thành các vùng, từ đó:
 - Nén ảnh
 - Phân đoạn ảnh
 - Nhận diện các vùng đặc trưng trong ảnh

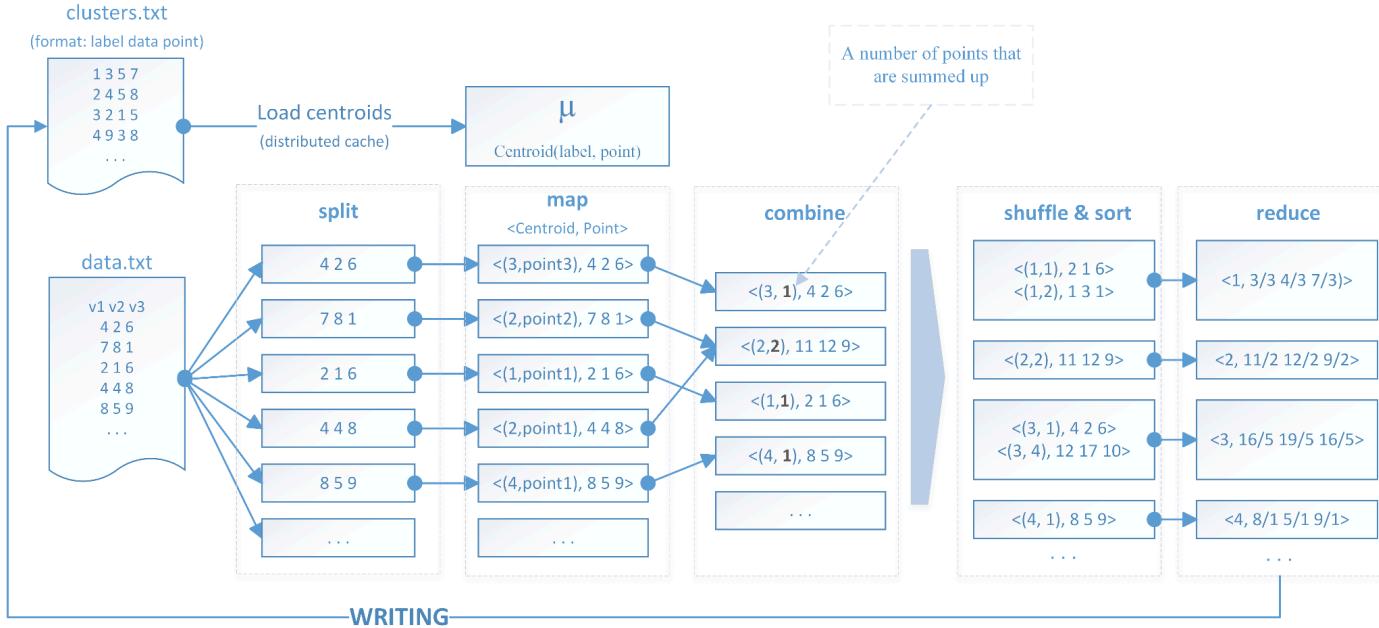
Ý tưởng:

- Chuyển đổi ảnh thành tập hợp điểm:
 - Mỗi pixel ảnh được biểu diễn dưới dạng một vector 3 chiều
- Khởi tạo tâm cụm:
 - Tâm cụm ban đầu được khởi tạo ngẫu nhiên bằng thuật toán Kmeans++
- Thực hiện MapReduce:
 - Mapper: tính khoảng cách từ từng điểm đến tất cả các tâm cụm và gán điểm đó vào cụm gần nhất
 - Reduce: tập hợp các điểm thuộc mỗi cụm, tính trung bình để cập nhật tâm cụm mới
- Lặp lại quá trình:
 - Quy trình MapReduce được lặp lại nhiều lần cho đến khi các tâm cụm hội tụ (không thay đổi hoặc thay đổi nhỏ hơn ngưỡng cho phép)

3.2 Lưu đồ của thuật toán MapReduce Kmeans



Hình 5. Lưu đồ thuật toán Mapreduce Kmeans



Hình 6. Ví dụ về cách thuật toán vận hành

3.3 Giải pháp MapReduce Kmeans trong phân cụm ảnh

Dữ liệu đầu vào: Dữ liệu đầu vào là 2 file .txt, 1 file là file lưu tất cả các điểm màu trong ảnh dưới dạng các vector giá trị RGB, file còn lại chứa các cụm được lấy ngẫu nhiên từ các điểm trong file ban đầu.

Triển khai:

1. **Biểu diễn Dữ liệu:**
 - Dữ liệu sẽ được biểu diễn dưới dạng danh sách các hàng, trong đó mỗi hàng là một vector giá trị RGB, đại diện cho một điểm ảnh.
2. **Lưu trữ phân tán dữ liệu:**
 - Dữ liệu đầu vào sẽ được chia thành các phần nhỏ (split) và phân phối tới các máy tính trong hệ thống để thực hiện song song việc xử lý.
3. **Xử lý trên từng máy tính trong mỗi vòng lặp:**
 - Trên mỗi máy tính, trong mỗi vòng lặp của thuật toán K-means, sẽ thực hiện các bước sau:
 - **Map:** Mỗi máy tính đọc một dòng dữ liệu (từng điểm RGB) và gán điểm đó vào cluster gần nhất bằng cách tính toán khoảng cách giữa điểm đó và các centroid (tâm cụm) hiện tại.
 - **Combiner:** Các kết quả map từ các máy tính sẽ được cộng gộp lại (combine) để giảm tải cho quá trình shuffle and sort.
 - **Shuffle and Sort:** Quá trình shuffle sẽ sắp xếp các kết quả theo key (tên cụm) và chuyển các điểm về đúng cluster của nó.

- **Reduce:** Trong bước reduce, sẽ tính toán lại centroid của từng cluster bằng cách tính giá trị trung bình của các điểm trong mỗi cluster. Sau đó, cập nhật centroid của từng cụm và kiểm tra xem thuật toán có hội tụ hay chưa (tức là centroid không thay đổi quá nhiều).

4. Quá trình lặp lại:

- Các bước Map, Combiner, Shuffle and Sort, Reduce sẽ được lặp lại cho đến khi các centroid hội tụ (nghĩa là sự thay đổi của các centroid giữa các vòng lặp liên tiếp là nhỏ hơn một ngưỡng xác định).

5. Đầu ra:

- Sau khi thuật toán K-means hội tụ, kết quả đầu ra sẽ là các centroid của từng cụm cùng với các điểm ảnh được gán vào các cụm tương ứng. Dữ liệu đầu ra này sẽ được lưu trữ dưới dạng các cặp key/value, trong đó key là số ID của cụm, còn value là vector biểu diễn cụm đó.
-

Mô hình cơ bản của MapReduce K-means trong phân cụm ảnh:

• Map:

- Đầu vào: Cặp (keyIn, valIn), trong đó:
 - **keyIn:** Là giá trị byte offset của dòng trong dữ liệu đầu vào.
 - **valIn:** Là giá trị của điểm ảnh (RGB).
- Xử lý: Dữ liệu được gán vào cluster gần nhất bằng cách tính khoảng cách giữa điểm RGB và các centroid hiện tại.
- Đầu ra: Cặp (keyOut, valOut), trong đó:
 - **keyOut:** Là label và point của cluster mà điểm RGB thuộc về.
 - **valOut:** Là giá trị điểm RGB.

• Combiner:

- Đầu vào: Cặp (keyOut, list(valOut)), trong đó:
 - **keyOut:** Là label và point của cluster.
 - **list(valOut):** Danh sách các điểm RGB thuộc cluster đó.
- Xử lý: Gộp và tính tổng point theo label của cluster.
- Đầu ra: (keyOut(label, sum(point), sum(valOut)).

• Shuffle and Sort:

- Quá trình shuffle sẽ sắp xếp các kết quả theo key (label của các cluster) và phân phối các điểm về đúng cluster.

• Reduce:

- Đầu vào: Cặp (keyOut, sum(valOut)).
- Xử lý: Tính toán centroid mới cho từng cluster bằng cách tính giá trị trung bình của các điểm trong cluster đó.
- Đầu ra: Cặp (keyFinal, valFinal), trong đó:
 - **keyFinal:** label của cluster.

- **valFinal**: Centroid (tâm cụm) mới.
-

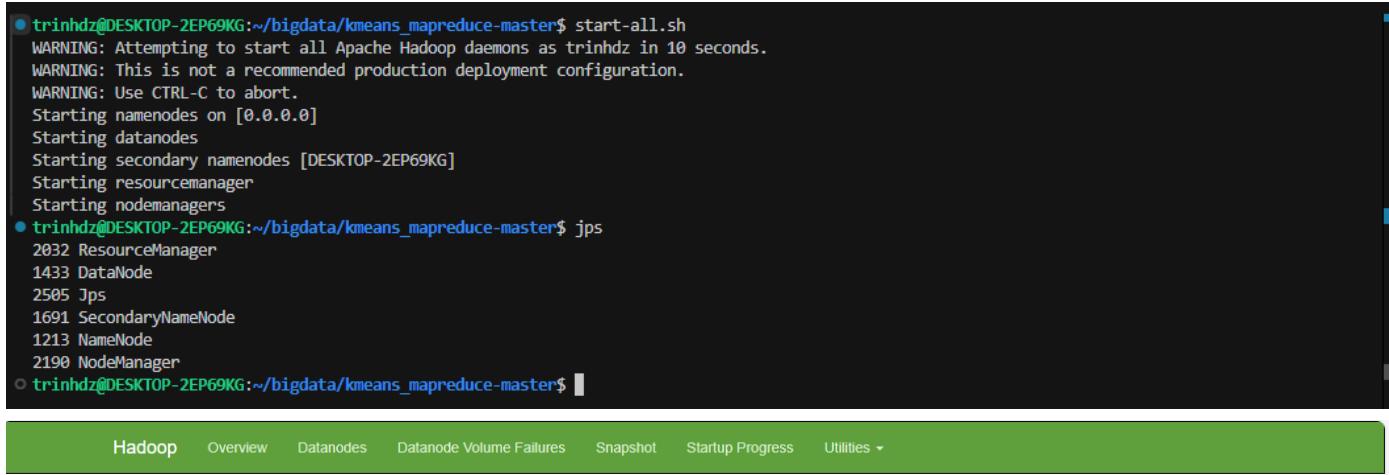
Cập nhật và hội tụ:

Sau mỗi vòng lặp, các centroid mới được tính toán lại. Quá trình này sẽ tiếp tục cho đến khi các centroid không thay đổi nhiều giữa các vòng lặp (đạt hội tụ) hoặc số vòng lặp đạt đến tối đa, khi đó thuật toán K-means kết thúc.

3.4 Demo chương trình cài đặt.

3.4.1 Demo cài đặt hadoop thành công.

```
trinhdz@DESKTOP-2EP69KG:~/bigdata/kmeans_mapreduce-master$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as trinhdz in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [0.0.0.0]
Starting datanodes
Starting secondary namenodes [DESKTOP-2EP69KG]
Starting resourcemanager
Starting nodemanagers
trinhdz@DESKTOP-2EP69KG:~/bigdata/kmeans_mapreduce-master$ jps
2032 ResourceManager
1433 DataNode
2505 Jps
1691 SecondaryNameNode
1213 NameNode
2198 NodeManager
trinhdz@DESKTOP-2EP69KG:~/bigdata/kmeans_mapreduce-master$
```



The screenshot shows a terminal window with a black background and white text. It displays the command `start-all.sh` being run, followed by a warning about running daemons as the user `trinhdz`. The terminal then lists the processes started: ResourceManager (pid 2032), DataNode (pid 1433), Jps (pid 2505), SecondaryNameNode (pid 1691), NameNode (pid 1213), and NodeManager (pid 2198). Below the terminal is a green navigation bar with links: Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, Utilities ▾.

Browse Directory

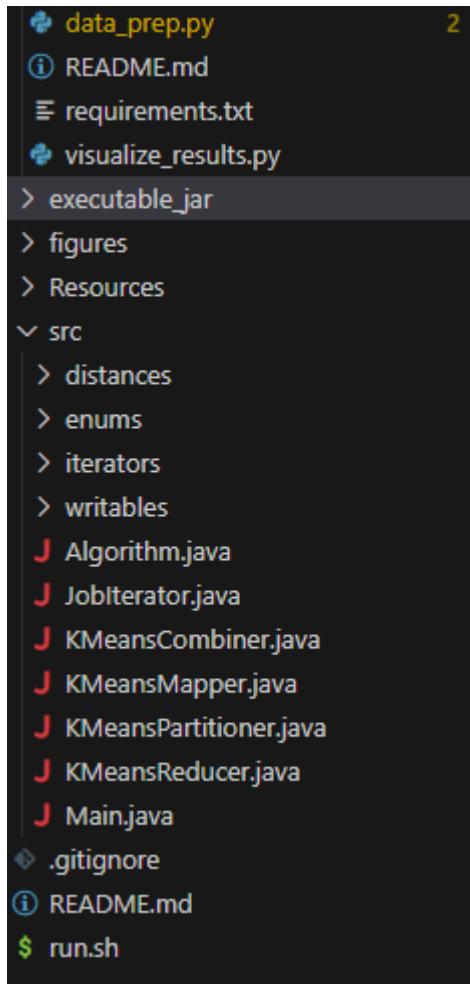
	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
<input type="checkbox"/>	drwxr-xr-X	trinhdz	supergroup	0 B	Nov 23 10:48	0	0 B	KMeans	
<input type="checkbox"/>	drwxr-xr-X	trinhdz	supergroup	0 B	Nov 23 10:49	0	0 B	tmp	
<input type="checkbox"/>	drwxr-xr-X	trinhdz	supergroup	0 B	Dec 02 07:55	0	0 B	user	

Show 25 entries Search:

Showing 1 to 3 of 3 entries Previous 1 Next

3.4.2 Demo Chương trình demo.

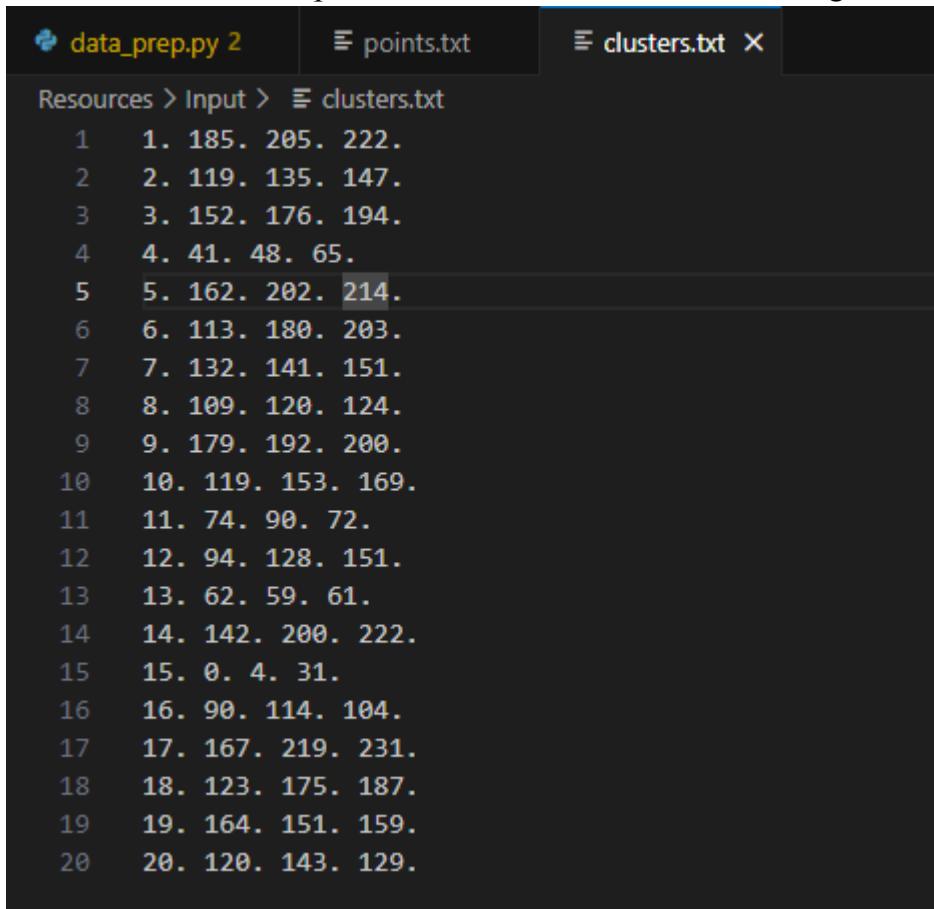
(1) Cấu trúc thư mục của project:



(2) Trước tiên ta chạy file data_prep.py trên ảnh đầu vào để có được file points.txt và clusters.txt:

```
● (venv) trinhdz@DESKTOP-2EP69KG:~/bigdata/kmeans_mapreduce-master$ python3 data_prep_scripts/data_prep.py
Points saved in: /home/trinhdz/bigdata/kmeans_mapreduce-master/Resources/Input/points.txt
Centroids saved in: /home/trinhdz/bigdata/kmeans_mapreduce-master/Resources/Input/clusters.txt
```

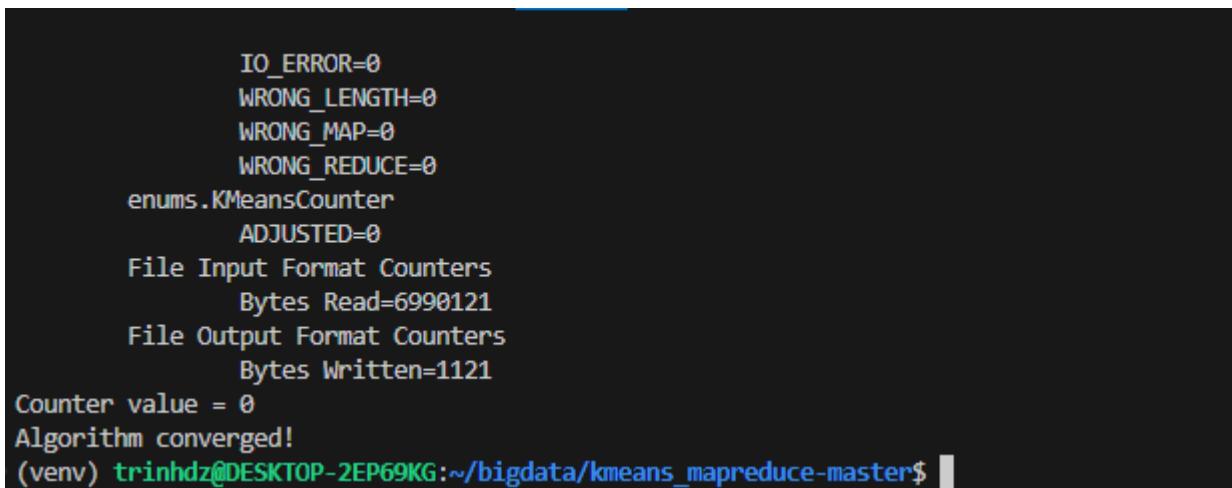
(3) Thu được file đầu vào points.txt và clusters.txt với định dạng:



```
data_prep.py 2      points.txt      clusters.txt X
Resources > Input > clusters.txt
1  1. 185. 205. 222.
2  2. 119. 135. 147.
3  3. 152. 176. 194.
4  4. 41. 48. 65.
5  5. 162. 202. 214.
6  6. 113. 180. 203.
7  7. 132. 141. 151.
8  8. 109. 120. 124.
9  9. 179. 192. 200.
10 10. 119. 153. 169.
11 11. 74. 90. 72.
12 12. 94. 128. 151.
13 13. 62. 59. 61.
14 14. 142. 200. 222.
15 15. 0. 4. 31.
16 16. 90. 114. 104.
17 17. 167. 219. 231.
18 18. 123. 175. 187.
19 19. 164. 151. 159.
20 20. 120. 143. 129.
```

Đây là file clusters được khởi tạo với 20 tâm cụm.

(4) Chạy Mapreduce Kmeans với clusters.txt và points.txt để phân cụm lại:



```
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
enums.KMeansCounter
ADJUSTED=0
File Input Format Counters
Bytes Read=6990121
File Output Format Counters
Bytes Written=1121
Counter value = 0
Algorithm converged!
(venv) trinhdz@DESKTOP-2EP69KG:~/bigdata/kmeans_mapreduce-master$
```

Sau khi chạy ta được 20 clusters mới:

```
$ run.sh      ≡ cluster_out.txt ×
Resources > Output > ≡ cluster_out.txt
 1  1 188.83657181447032 210.70205973539615 222.20489916198545
 2  2 114.84856976065382 135.5415061295972 148.99614711033274
 3  3 148.9986457912618 179.33391936518495 194.06913591979284
 4  4 37.235395189003434 47.012027491408936 61.76052405498282
 5  5 161.70850700758106 198.48658628696114 213.5855876017568
 6  6 115.30189972581277 178.43272620446533 205.40888170779476
 7  7 134.58301268348296 145.41299700726807 151.81986603961806
 8  8 106.92742703732571 118.41326751537196 123.92569498571058
 9  9 177.76208201501333 191.71648365283207 200.13660897077983
10 10 115.50293931446143 153.9484790931838 171.53405082753008
11 11 72.42077315208157 83.97313084112149 87.30140186915888
12 12 90.5454057247194 129.35497556522205 151.5883679716449
13 13 59.60784844384303 63.61948579161029 66.89526387009472
14 14 140.54063383995782 195.5380927266625 215.5459078486839
15 15 15.007504690431519 15.51106941838649 23.38611632270169
16 16 86.14774968621123 102.9896001434463 109.36578805809575
17 17 173.0417855686914 217.19894007337953 231.6523644516918
18 18 125.88694684107988 171.91636515446703 187.85318675201782
19 19 154.79323308270676 160.77923976608187 163.86800334168754
20 20 122.56463692608271 136.80039075219798 130.63725170954086
21
```

(5) Chạy file visualize_results.py để tạo ra ảnh mới với 20 cụm màu:



Hình ảnh gốc và hình ảnh sau khi được phân cụm

CHƯƠNG 4: CẢI TIẾN THUẬT TOÁN

4.1 Bổ sung hàm khoảng cách:

- Với việc sử dụng khoảng cách cũ Euclidean Distance nó sẽ có một số hạn chế:
 - Nếu cụm màu không tách biệt rõ, các màu sắc có thể không được nhóm một cách tối ưu
 - Thời gian tính toán có thể được kéo dài
- => Từ đó đòi hỏi việc phải kết hợp một hàm khoảng cách mới phù hợp
- Hàm khoảng cách Cosine Distance được lựa chọn:
 - Giảm nhiễu nếu dữ liệu có độ chênh lệch lớn về không gian
 - Cụm màu có thể được phân biệt rõ ràng hơn nếu các ảnh có nhiều màu với độ sáng khác nhau
- Triển khai:
 - Lấy dữ liệu đặc trưng: mỗi điểm ảnh được biểu diễn dưới dạng 1 vector đặc trưng
 - Kiểm tra kích thước vector: đảm bảo 2 vector tính toán có cùng chiều dài
 - Tính tích vô hướng giữa 2 vector
 - Tính khoảng cách cosine
 - Sau khi tính phân nhóm các điểm ảnh, tính toán giá trị kì vọng để cập nhật các tâm cụm

4.2 Cải tiến thuật toán phân cụm Kmeans++:

- Các tâm cụm hiện tại được khởi tạo ngẫu nhiên:
 - Dẫn đến hội tụ chậm hoặc kém hiệu quả
 - Các tâm cụ có thể bị lệch do tâm cụm ban đầu không phản ánh đúng phân phối của dữ liệu
- Cải thiện tâm cụm bằng thuật toán Kmeans++:
 - Số vòng lặp để hội tụ giảm dần đến hội tụ nhanh hơn
 - Việc khởi tạo bằng Kmeans++ các cụm sẽ đồng đều hơn, phản ánh rõ cấu trúc của dữ liệu
- Triển khai:
 - Một tâm cụm được chọn ngẫu nhiên trong tập khởi tạo
 - Tính khoảng cách đến centroid đã chọn
 - Chọn centroid tiếp theo: việc chọn này không ngẫu nhiên, mà có xác suất tỷ lệ bình phương khoảng cách từ điểm ảnh đến centroid gần nhất
 - Lặp lại bước 2 và bước 3 đến khi đủ số lượng centroid
 - Khi số lượng centroid đã đủ phân nhóm các điểm ảnh và cụm dựa trên khoảng cách gần nhất đến các centroid

4.3 Thủ nghiệm với HOG:

- Đặc trưng HOG thường được dùng để biểu diễn các đường viền và cạnh trong ảnh, giúp mô tả hình dạng vật thể.
- việc sử dụng HOG có thể giúp giảm kích thước ảnh và giúp giảm độ phức tạp tính toán.

```
Bytes Written=218
Counter value = 0
Algorithm converged!
1 0.02242794304369558
2 0.3372376365337106
3 0.1963627227380824
4 0.08939259054791666
5 0.24110169961848657
6 0.07896372169960783
7 0.05569594682727643
8 0.11178326005821851
9 0.15205369768584812
10 0.1757937823339315
○ (env) phamphong@PhongLap:~/Kmeans-mapreduce$
```

10 tâm cụm được lấy khi sử dụng HOG

4.4 Thủ nghiệm với CMYK:



Qua kết quả ta thấy rằng việc sử dụng không gian màu CMYK thay vì RGB mang lại nhiều cải thiện đáng kể trong việc xử lý và phân cụm ảnh. CMYK giúp biểu diễn các đặc điểm màu sắc một cách tối ưu hơn trong trường hợp hình ảnh có sự tương đồng cao về màu giữa nền và vật thể, hoặc chứa nhiều nhiễu về màu sắc. Điều này là do CMYK chia tách thông tin màu sắc thành các thành phần dựa trên cường độ của mực in, từ đó giảm bớt sự phụ thuộc vào ánh sáng như trong RGB.

Không gian màu CMYK cũng hữu ích khi làm việc với các hình ảnh có tông màu nhạt hoặc cần phân biệt giữa các mức sắc độ rất nhỏ, giúp thuật toán phân cụm như k-means nhận diện các nhóm dữ liệu hiệu quả hơn. Bằng cách chuyển

đổi từ RGB sang CMYK, dữ liệu đầu vào trở nên phù hợp hơn để phân tích và xử lý, đặc biệt khi cần tìm kiếm sự khác biệt tinh tế giữa các nhóm.

Nhìn chung, việc tích hợp không gian màu CMYK vào quy trình phân cụm không chỉ giảm thiểu các vấn đề liên quan đến nhiều màu mà còn tăng độ chính xác trong việc xác định các cụm, đặc biệt hữu ích cho các bài toán đòi hỏi sự khác biệt rõ ràng trong đặc trưng màu sắc.

4.5 Kết quả và đánh giá:



- Qua kết quả ta thấy việc sử dụng đặc trưng HOG mang lại nhiều cải thiện đáng kể trong việc xử lý và phân cụm ảnh. HOG giúp biểu diễn các đặc điểm quan trọng như biên và cạnh trong ảnh, thay vì dựa vào giá trị màu sắc như trong phương pháp truyền thống. Điều này sẽ đặc biệt hữu ích trong việc phân cụm các hình ảnh phức tạp, có nhiều điểm nhiều màu hoặc màu giữa nền và vật thể có độ tương tự cao.
- Ngoài ra, việc giảm kích thước đặc trưng của dữ liệu thông qua HOG không chỉ giúp giảm tải bộ nhớ mà còn tăng hiệu quả tính toán. Đặc trưng này giữ lại các thông tin quan trọng về cấu trúc của ảnh, giúp cải thiện độ chính xác của thuật toán phân cụm.

=> Nhìn chung, việc tích hợp HOG vào quy trình phân cụm giúp giải quyết một số trường hợp ảnh có nhiều điểm màu nhiều hoặc có ít đặc trưng màu, tạo điều kiện cho thuật toán hoạt động hiệu quả hơn, đặc biệt trong các bài toán yêu cầu mô tả hình dạng và cấu trúc tổng thể của đối tượng trong ảnh.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Ứng dụng thực tế của thuật toán: Phân loại và phân cụm ảnh trong phân tích dữ liệu

- Phân tích ảnh y tế: Ví dụ, trong y tế, các bức ảnh chụp X-quang, MRI hoặc CT có thể được phân cụm để phát hiện các vùng bất thường hoặc phân loại các loại bệnh. K-means MapReduce giúp phân tích hàng triệu bức ảnh y tế trên một hệ thống phân tán mà không gặp phải vấn đề về bộ nhớ.
- Nhận dạng đối tượng: K-means có thể được sử dụng để phân cụm các đối tượng trong ảnh, giúp nhận dạng các nhóm đối tượng tương tự. MapReduce giúp xử lý hiệu quả các tập dữ liệu ảnh lớn, chẳng hạn như trong phân tích dữ liệu ảnh vệ tinh hoặc ảnh từ camera giám sát.

Tìm kiếm ảnh và tổ chức thư viện ảnh

- Tổ chức thư viện ảnh: Khi người dùng có một bộ sưu tập ảnh lớn, K-means có thể được dùng để phân loại các ảnh thành các nhóm có nội dung tương tự, ví dụ như ảnh của các địa điểm, sự kiện, hoặc các đối tượng cụ thể. Công nghệ MapReduce giúp triển khai quá trình phân cụm trên bộ sưu tập ảnh lớn.
- Tìm kiếm ảnh theo nội dung (Content-Based Image Retrieval - CBIR): Khi tìm kiếm ảnh tương tự từ một bộ sưu tập lớn, K-means MapReduce có thể giúp phân cụm các ảnh có đặc điểm tương đồng về màu sắc, kết cấu, hoặc hình dạng, từ đó giúp việc tìm kiếm ảnh nhanh chóng và chính xác hơn.

Phân tích và tối ưu hóa ảnh trong ứng dụng trực tuyến

- Chẩn đoán hình ảnh: Các ứng dụng trong lĩnh vực học máy và AI (chẳng hạn trong các hệ thống nhận diện hình ảnh) có thể sử dụng K-means MapReduce để phân cụm các đối tượng trong ảnh, giúp cải thiện độ chính xác của mô hình nhận diện.
- Phân tích ảnh từ mạng xã hội: Trên các nền tảng như Facebook, Instagram, K-means MapReduce có thể giúp phân loại và nhóm ảnh của người dùng theo các chủ đề hoặc các yếu tố tương tự, tạo ra trải nghiệm tìm kiếm và gợi ý ảnh tốt hơn.

Phân tích ảnh vệ tinh và địa lý

- Phân tích ảnh vệ tinh: K-means MapReduce có thể được sử dụng trong các ứng dụng phân tích ảnh vệ tinh, giúp phân cụm các khu vực có đặc điểm tương đồng (ví dụ, đất nông nghiệp, khu đô thị, rừng,...) trên ảnh vệ tinh.

Các hệ thống phân tán sẽ giúp xử lý dữ liệu vệ tinh quy mô lớn, giúp nâng cao độ chính xác của các dự báo hoặc phân tích địa lý.

5.2 Kết luận

K-means MapReduce không chỉ là công cụ mạnh mẽ trong xử lý ảnh mà còn là cầu nối giữa thuật toán truyền thống và công nghệ xử lý dữ liệu phân tán hiện đại, giúp giải quyết các bài toán phân cụm ảnh quy mô lớn một cách hiệu quả và chính xác.

5.3 Hướng phát triển

Kết hợp học sâu (Deep Learning) trong phân cụm ảnh: Kết hợp K-means và MapReduce với các phương pháp học sâu để cải thiện chất lượng phân cụm, đặc biệt trong các bài toán phân cụm ảnh phức tạp, như phân cụm ảnh trong các không gian đặc trưng cao.

Phân cụm ảnh dựa trên đặc trưng học được (feature learning): Các nghiên cứu có thể tích hợp K-means với các kỹ thuật học đặc trưng để tự động học các đặc trưng từ ảnh, thay vì dựa vào các đặc trưng truyền thống, từ đó cải thiện chất lượng phân cụm.

Phân tích ảnh trong thời gian thực: Các ứng dụng như giám sát video trực tuyến hoặc phân tích ảnh trong các hệ thống giao thông thông minh cần xử lý ảnh thời gian thực. Hướng nghiên cứu có thể tập trung vào việc phát triển các kỹ thuật phân cụm hiệu quả, dễ mở rộng và có thể hoạt động trong môi trường phân tán, cho phép xử lý hàng triệu video và hình ảnh mỗi giây.

Phân tích ảnh không gian (geospatial analysis): Với sự phát triển của các hệ thống GIS (Hệ thống Thông tin Địa lý), K-means MapReduce có thể được sử dụng để phân tích và phân loại các loại dữ liệu ảnh vệ tinh quy mô lớn, chẳng hạn như theo dõi sự thay đổi đất đai hoặc phân tích môi trường.

TÀI LIỆU THAM KHẢO

- [1] https://github.com/markomih/kmeans_mapreduce/tree/master
- [2] http://vap.ac.vn/Portals/0/TuyenTap/2021/12/22/1ecec417207345d595e011cb434f7fe8/1_FAIR2021_paper_25.pdf
- [3] <https://github.com/free-dino/KMeans-Mapreduce-for-Image-Segmentation>
<https://www.slideshare.net/A13Superman/hadoop-h-thng-tnh-ton-v-x-l-d-liu-ln>
- [4] https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
- [5] https://www.geeksforgeeks.org/hashmap-class-methods-java-examples-set-1-put-get-i_sempty-size/
- [6] https://www.academia.edu/11449110/Luan_Van_Hadoop_Final
- [7] <https://pdfs.semanticscholar.org/a77d/d6aad21d0c75a3a1cf0c7a21cbc0378cf865.pdf>

NHIỆM VỤ CỦA CÁC THÀNH VIÊN

Họ và tên	Công việc
Nguyễn Viết Vũ	+ Tìm hiểu tổng quan về map reduce.

	<ul style="list-style-type: none"> + Tìm hiểu bài toán phân cụm ảnh, chạy thử code + Phân tích bài toán và phân chia công việc cho các thành viên + Theo dõi tiến độ + Tổng hợp kết quả và sửa đổi + Tạo bản python mapreduce + Làm báo cáo + Tạo slide
Nguyễn Tông Quân	<ul style="list-style-type: none"> + Tìm hiểu tổng quan về hadoop. + Lên ý tưởng cải thiện bài toán + Thực hiện tìm hiểu và cải tiến hiệu suất, tốc độ của các thuật toán Kmeans + Thay đổi hàm khoảng cách, sửa đổi thuật toán phân cụm Kmeans++ + Làm báo cáo.
Nguyễn Xuân Trình	<ul style="list-style-type: none"> + Tìm hiểu về giải thuật Kmeans. + Sửa đổi một số lỗi còn tồn tại + Thực nghiệm Kmeans++ + Ví dụ minh họa cho thuật toán Kmeans. + Vẽ lưu đồ thuật toán phân cụm Kmeans + Cài đặt chương trình demo. + Làm báo cáo

Phạm Đăng Phong	<ul style="list-style-type: none">+ Tìm hiểu tổng quan, đặc trưng, ứng dụng của big data.+ Tìm hiểu về bước tiền xử lý dữ liệu+ Chuyển dữ liệu ảnh RGB sang CMYK và HOG để thử nghiệm+ Làm báo cáo
-----------------------	---