



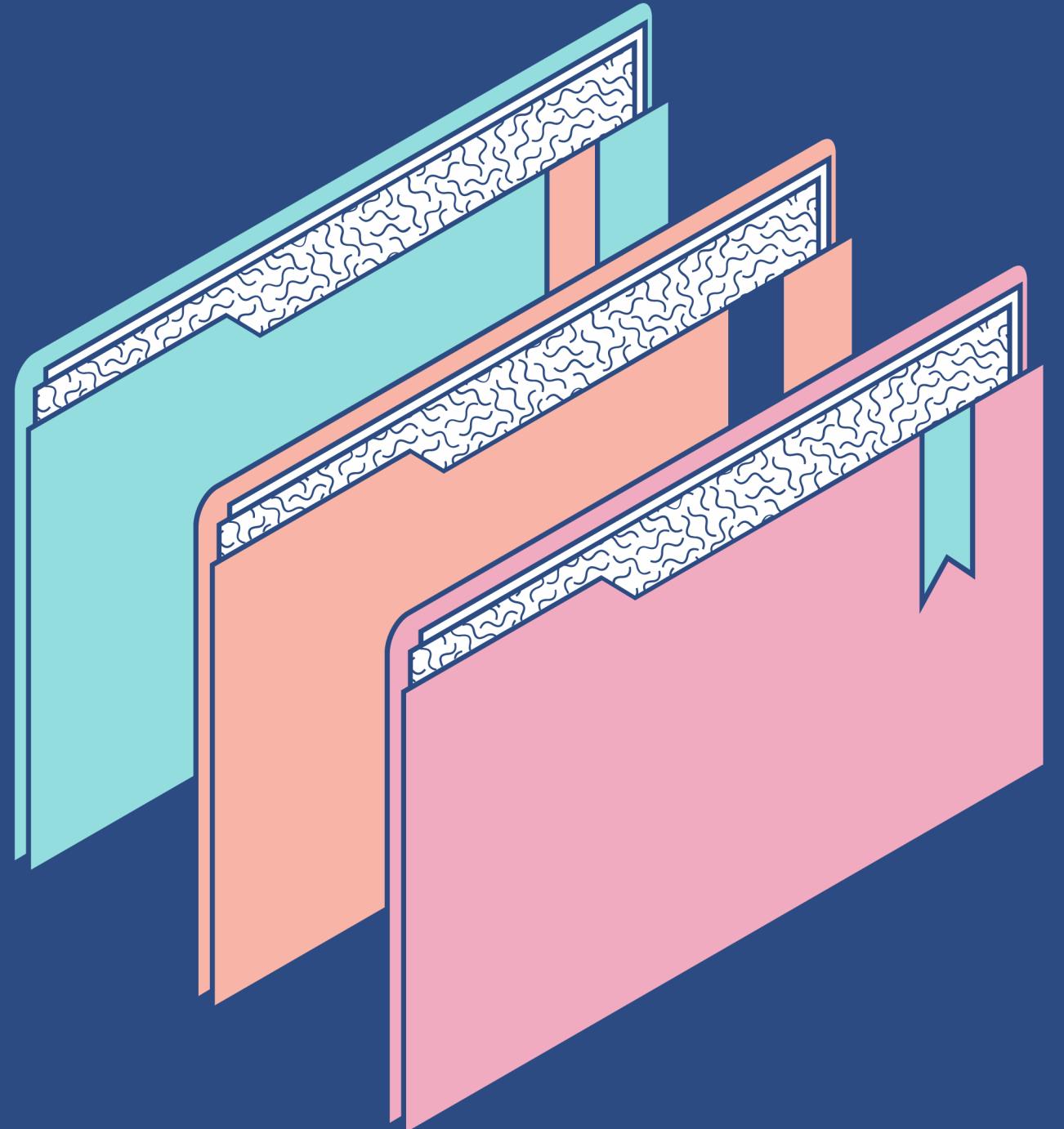
BÁO CÁO MÔN HỌC KỸ THUẬT VÀ CÔNG NGHỆ DỮ LIỆU
LỚN

K-MEANS VÀ LẬP TRÌNH MAPREDUCE HÓA TRONG PHÂN CỤM ẢNH

Nhóm sinh viên thực hiện:

1. Vũ Việt Hùng - 22022585
2. Hồ Minh Hoàng - 22022567
3. Nguyễn Trọng Huy - 22022545
4. Nguyễn Đức Anh - 22022661
5. Hà Kim Dương - 22022621

Mục lục



BÁO CÁO GỒM 4 CHƯƠNG:

- Tổng quan về dữ liệu lớn.
- Thuật toán phân cụm K Means.
- Ứng dụng của Kmeans và MapReduce trong phân cụm ảnh.
- Kết luận và hướng phát triển.

TỔNG QUAN VỀ DỮ LIỆU LỚN

Big data là tập dữ liệu lớn mà chúng ta không thể xử lý bằng phương pháp truyền thống.



Tổng quan về dữ liệu lớn

Đặc trưng cơ bản của bigdata

1	2	3	4	5
KHỐI LƯỢNG LỚN (VOLUME)	TỐC ĐỘ (VELOCITY)	ĐA DẠNG (VARIETY)	ĐỘ TIN CẬY/CHÍNH XÁC (VERACITY)	GIÁ TRỊ (VALUE)
Khối lượng dữ liệu rất lớn và đang ngày càng tăng lên, tính đến 2014 thì có thể trong khoảng vài trăm terabyte.	Khối lượng dữ liệu gia tăng rất nhanh.	Ngày nay hơn 80% dữ liệu được sinh ra là phi cấu trúc(tài liệu, blog, hình ảnh,...)	Bài toán phân tích và loại bỏ dữ liệu thiếu chính xác và nhiễu đang là tính chất quan trọng của bigdata.	Giá trị thông tin mang lại.

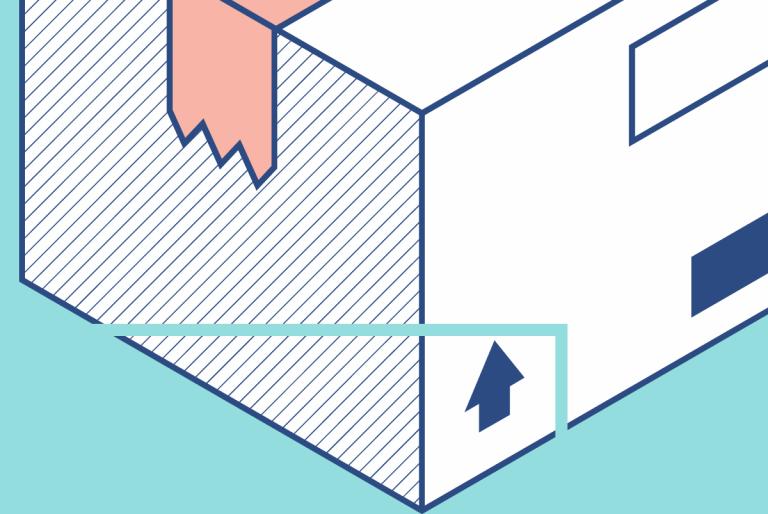
Tổng quan về dữ liệu lớn

Tổng quan về hadoop

1. Apache Hadoop là một framework hỗ trợ cho việc lưu trữ và xử lý dữ liệu phân tán trên nhiều máy.

2. Thành phần:

- Hadoop Common
- HDFS: Hệ thống file phân tán
- Hadoop YARN Framework dùng cho lập lịch job và quản lý tài nguyên của hệ thống
- MapReduce : Hệ thống xử lý dữ liệu của Hadoop



Tổng quan về dữ liệu lớn

Tổng quan về MapReduce

MapReduce gồm 2 thủ tục chính:

- Thủ tục Map: Có vai trò lọc và phân loại dữ liệu
- Thủ tục Reduce: Tổng hợp dữ liệu

B1: Phân rã từ nghiệp vụ chính

Phân rã từ nghiệp vụ chính (do người dùng muốn thể hiện) thành các công việc con để chia từng công việc con này về các máy tính trong hệ thống thực hiện xử lý một cách song song

B2: Thu thập lại kết quả

Mô hình MapReduce:

- Hàm Map : Hàm Map tiếp nhận mảnh dữ liệu input, rút trích thông tin cần thiết các từng phần tử (ví dụ: lọc dữ liệu, hoặc trích dữ liệu) tạo kết quả trung gian
- Shuffle: Dữ liệu đầu ra từ bước Map được sắp xếp và gom nhóm để chuẩn bị cho bước Reduce
- Hàm Reduce: tổng hợp kết quả trung gian, tính toán để cho kết quả cuối cùng

THUẬT TOÁN PHÂN CỤM K-MEANS





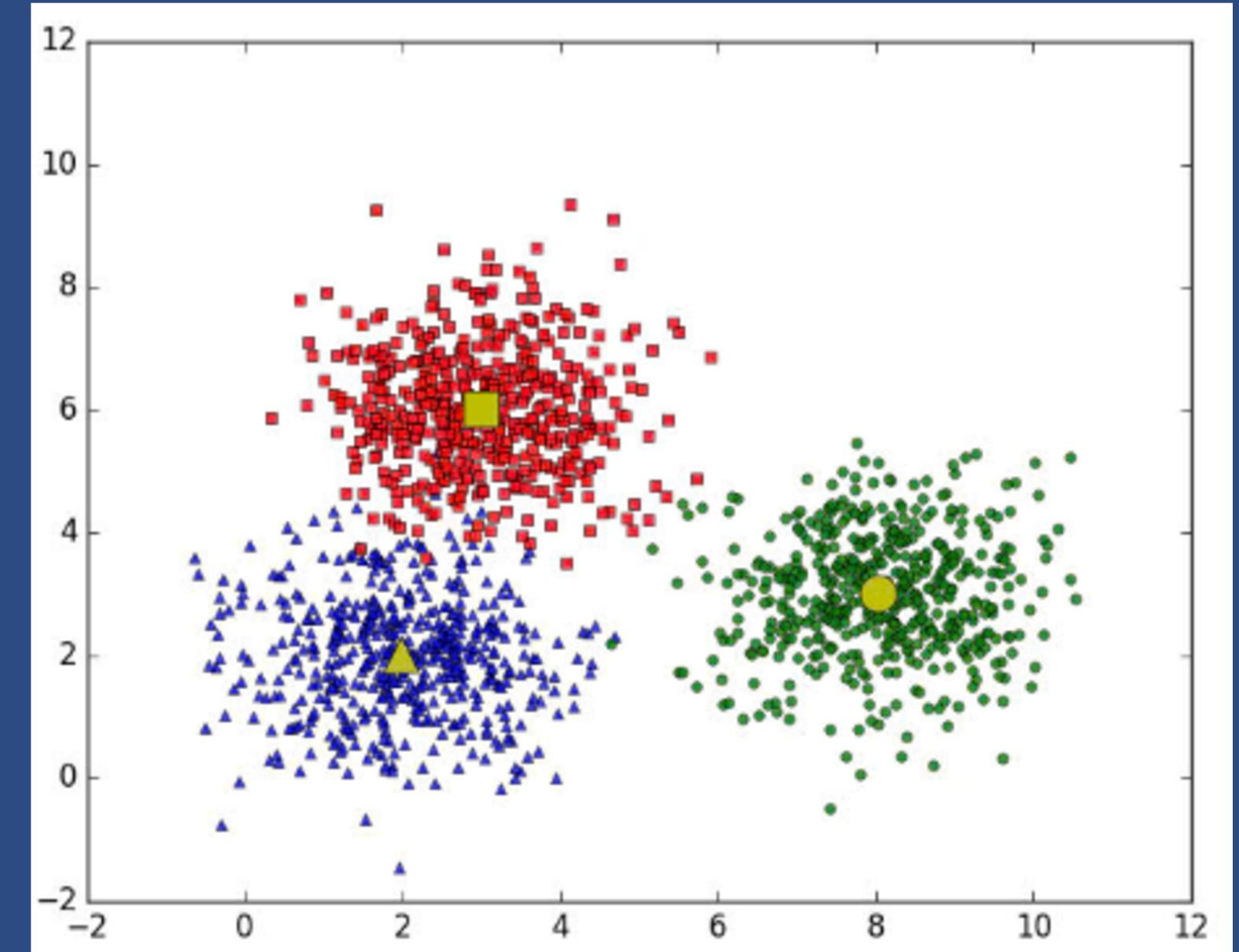
Giới thiệu thuật toán K-means clustering

- Thuật toán K-means được giới thiệu năm 1957 bởi Stuart Lloyd và là phương pháp phổ biến nhất cho việc phân cụm ,dựa trên việc phân vùng dữ liệu
- Một trong những thuật toán cơ bản của Unsupervised Learning

Thuật toán phân cụm K-Means

Giới thiệu

- Chúng ta không biết nhãn (label) của từng điểm dữ liệu. Mục đích là làm thế nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong cùng một cụm có tính chất giống nhau.
- Biểu diễn dữ liệu: $D = \{x_1, x_2, x_3, \dots, x_r\}$ với x_i là vector n chiều trong không gian Euclidean.
K-means phân cụm D thành K cụm dữ liệu:
 - Mỗi cụm có một điểm trung tâm gọi là centroid
 - K là một hằng số cho trước
- Ý tưởng: Chia 1 bộ dữ liệu thành các cụm khác nhau



Thuật toán phân cụm K-Means

Triển khai

Các bước thực hiện:

1

Chọn ngẫu nhiên
K điểm trong dữ
liệu X làm các
center ban đầu
 $M^{(0)}$

2

Với mỗi điểm
dữ liệu x_i ,
tính khoảng
cách đến tất
cả các center
hiện tại. Gán
điểm x_i vào
cụm có center
gần nhất

$$j = \arg \min_j \|x_i - m_j\|_2^2$$

3

Cập nhật center
với mỗi cụm k,
tính center mới
bằng trung bình
cộng tất cả các
điểm dữ liệu
thuộc về cụm đó.
Trong đó, C_k là
tập hợp các điểm
dữ liệu thuộc
cụm k, và $|C_k|$
là số điểm trong
cụm

$$m_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

4

Kiểm tra tính hội
tụ. Nếu việc gán
dữ liệu Y không
thay đổi so với
vòng lặp trước,
hoặc centers M
không thay đổi
đáng kể, dừng
thuật toán

5

Nếu chưa hội tụ
quay lại bước 2

Thuật toán phân cụm K-Means

Ví dụ về một vòng lặp của thuật toán K-Means

Giả sử ta có 6 điểm dữ liệu tương ứng với 6 điểm ảnh:

Point 1: (255, 0, 0)

Point 2: (254, 1, 2)

Point 3: (0, 255, 0)

Point 4: (1, 254, 1)

Point 5: (0, 0, 255)

Point 6: (2, 3, 250)

Với $K = 3$ tương ứng với 3 tâm được khởi tạo:

Centroid 1: (200, 0, 0)

Centroid 2: (0, 200, 0)

Centroid 3: (0, 0, 200)

Thuật toán phân cụm K-Means

Ví dụ về một vòng lặp của thuật toán K-Means

Sử dụng khoảng cách Euclidean để tính khoảng cách giữa các điểm ảnh và tâm

	CENTROID 1: (200, 0, 0)	CENTROID 2: (0, 200, 0)	CENTROID 3: (0, 0, 200)
Point 1: (255, 0, 0)	55.0	320.2	320.2
Point 2: (254, 1, 2)	54.2	319.1	318.1
Point 3: (0, 255, 0)	320.2	55.0	320.2
Point 4: (1, 254, 1)	318.4	54.2	318.1
Point 5: (0, 0, 255)	320.2	320.2	55.0
Point 6: (2, 3, 250)	318.3	319.1	54.2

Thuật toán phân cụm K-Means

Ví dụ về một vòng lặp của thuật toán K-Means

Phân các điểm dữ liệu về các cụm:

Centroid 1: (Point 1, Point 2)

Centroid 2: (Point 3, Point 4)

Centroid 3: (Point 5, Point 6)

Tính toán Centroid mới:

Centroid 1 mới : (254.5, 0.5, 1.0)

Centroid 2 mới : (0.5, 254.5, 0.5)

Centroid 3 mới : (1.0, 1.5, 252.5)



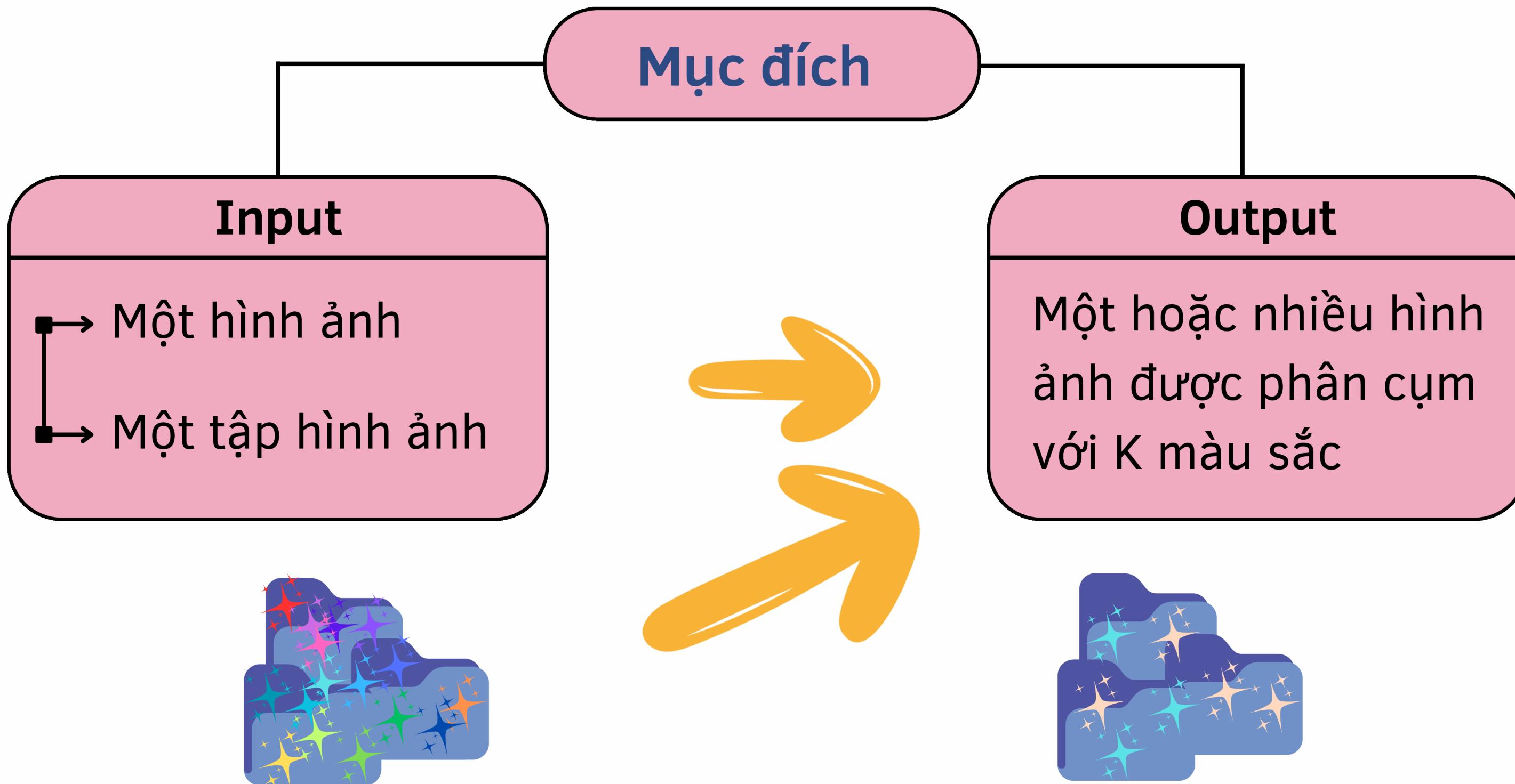
Ứng dụng Map Reduce trong Phân cụm ảnh

- Giới thiệu bài toán
- Cách tiếp cận
- Triển khai
- Thực nghiệm

```
3   require File.expand_path("../../config/environment", __FILE__)
4   # Prevent database truncation if the database needs cleaning
5   abort("The Rails environment is running in production mode!
6       Run `rake db:reset` to perform the reset operations from
7       the script.
8
9   require 'spec_helper'
10  require 'rspec/rails'
11
12  require 'capybara/rspec'
13  require 'capybara/rails'
14
15  Capybara.javascript_driver = :webkit
16  Category.delete_all; Category.create!
17  Shoulda::Matchers.configure do |config|
18    config.integrate do |with|
19      with.test_framework :rspec
20      with.library :rails
21    end
22  end
23
24  # Add additional requires below this line to avoid
25  # Requires supporting files via `require` statements in
26  # spec/support/ and its subdirectories.
27  #
28  # run as spec files by default. This means you will
29  # in _spec.rb will both be required as
30  # run twice. It is recommended that you name
31  # end with _spec.rb. You can configure the
32  # option on the command line via
33  #
34  # option found for 'mongoid'
35
36  Mongoid.configure do |config|
37    config.master = "localhost:27017"
38    config.buffer
```

Ứng dụng Map Reduce trong Phân cụm ảnh

1. Giới thiệu bài toán

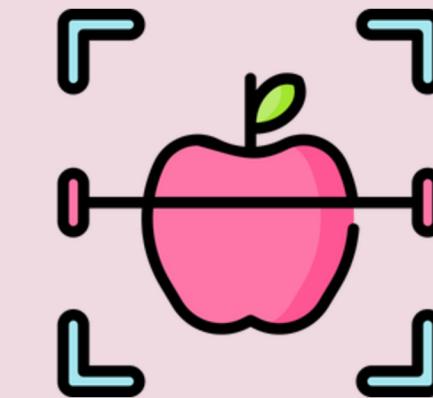


Ứng dụng Map Reduce trong Phân cụm ảnh

1. Giới thiệu bài toán

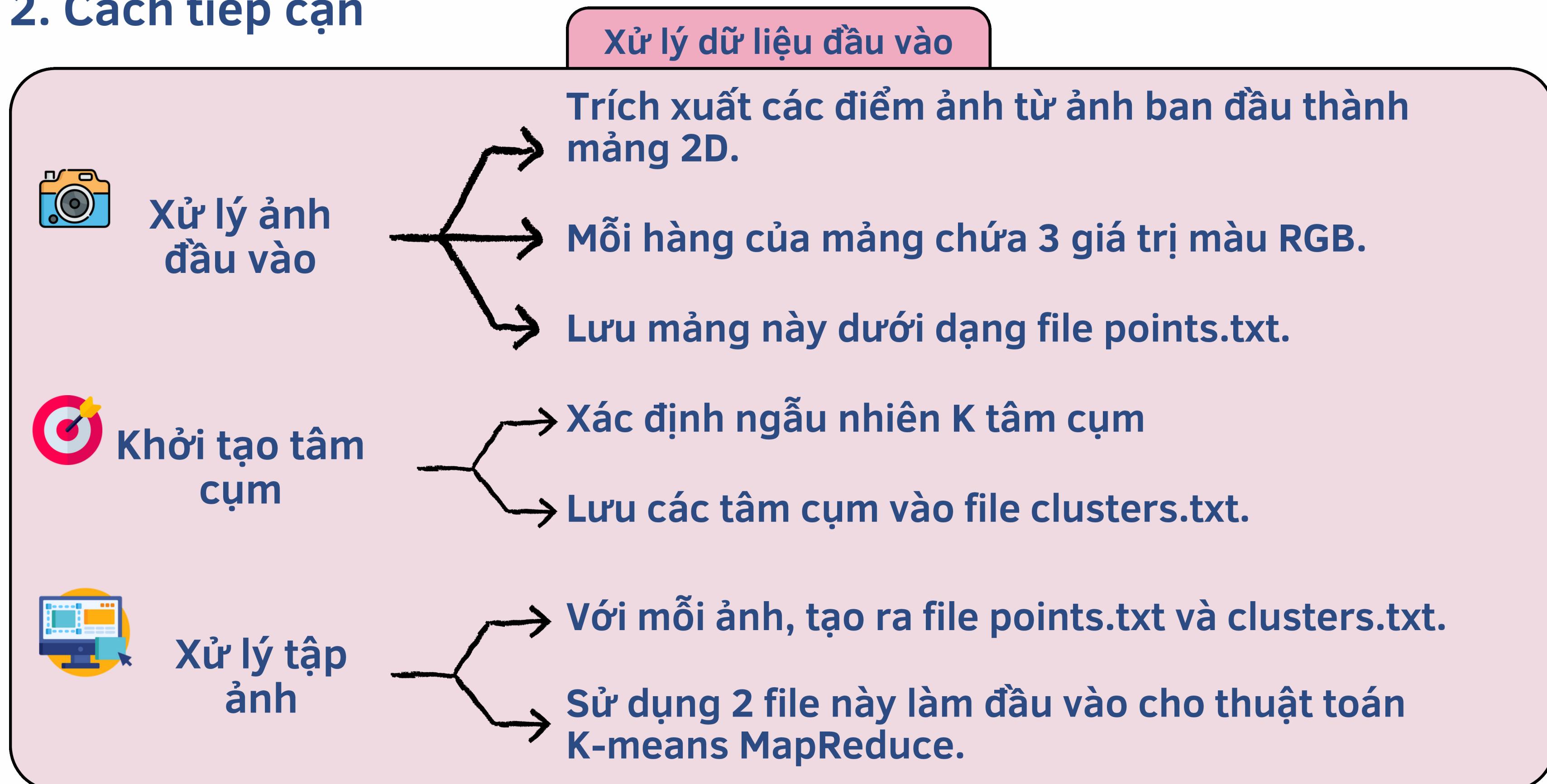
Ứng dụng

- Giảm số lượng màu sắc.
- Nhận diện đối tượng.
- Tách nền trong ảnh.
- Hỗ trợ y tế (phân vùng tế bào trong ảnh X-quang,...).
- Phân loại cảnh quan.
- Phân tích vệ tinh.
- Tiền xử lý cho các thuật toán thị giác máy tính.
- Giảm dữ liệu đầu vào và trích xuất các đặc trưng quan trọng giúp giảm tài nguyên tính toán và giảm bộ nhớ cần thiết.



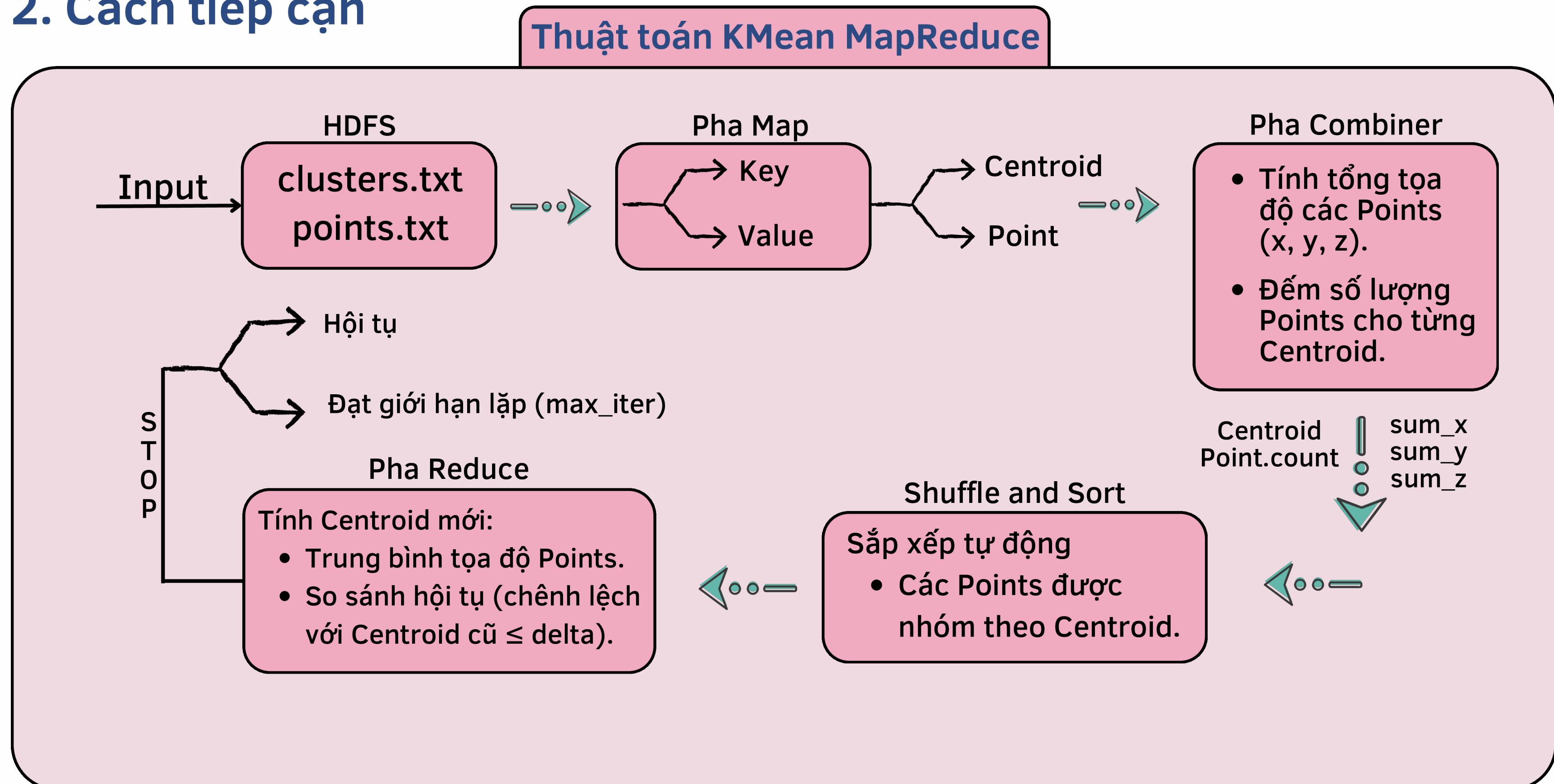
Ứng dụng Map Reduce trong Phân cụm ảnh

2. Cách tiếp cận



Ứng dụng Map Reduce trong Phân cụm ảnh

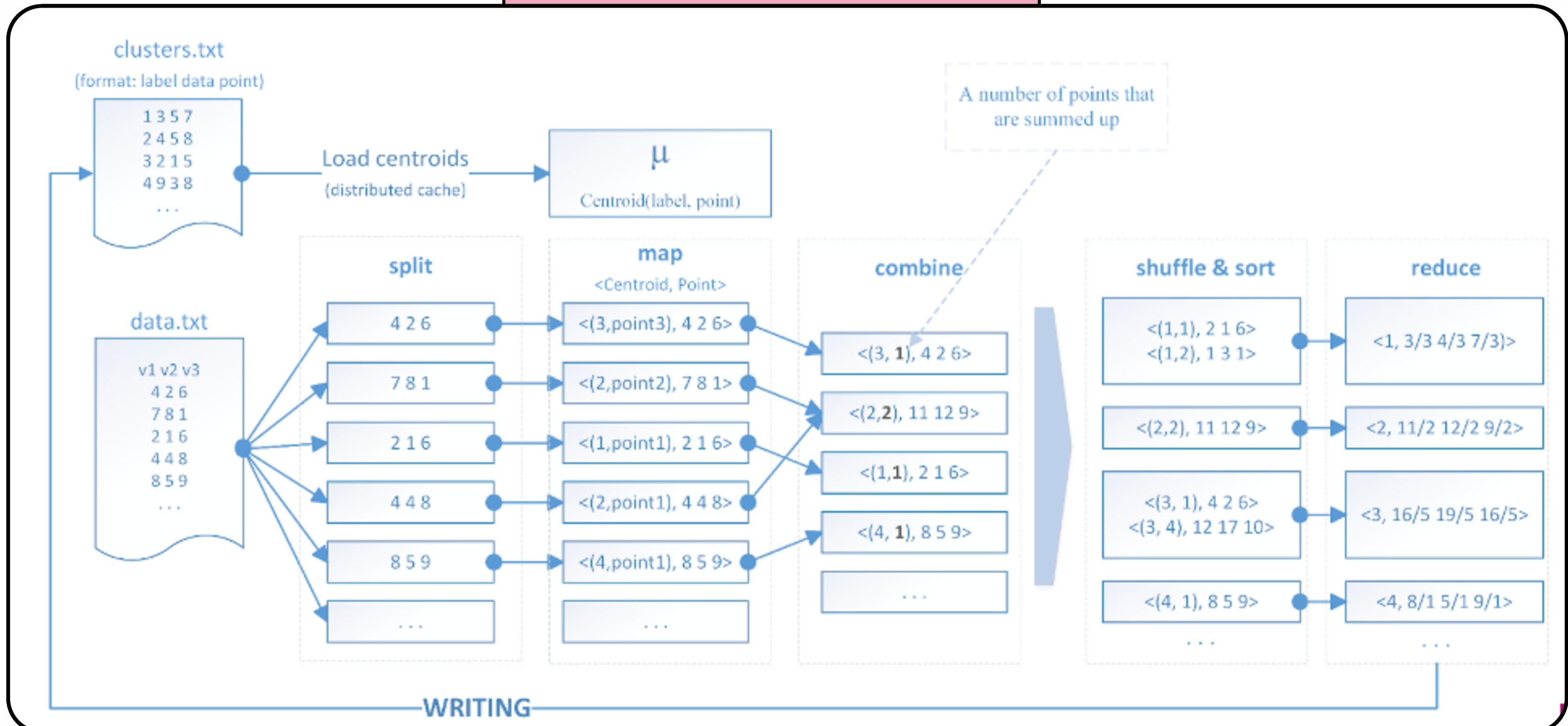
2. Cách tiếp cận



Ứng dụng Map Reduce trong Phân cụm ảnh

2. Cách tiếp cận

Thuật toán KMean MapReduce



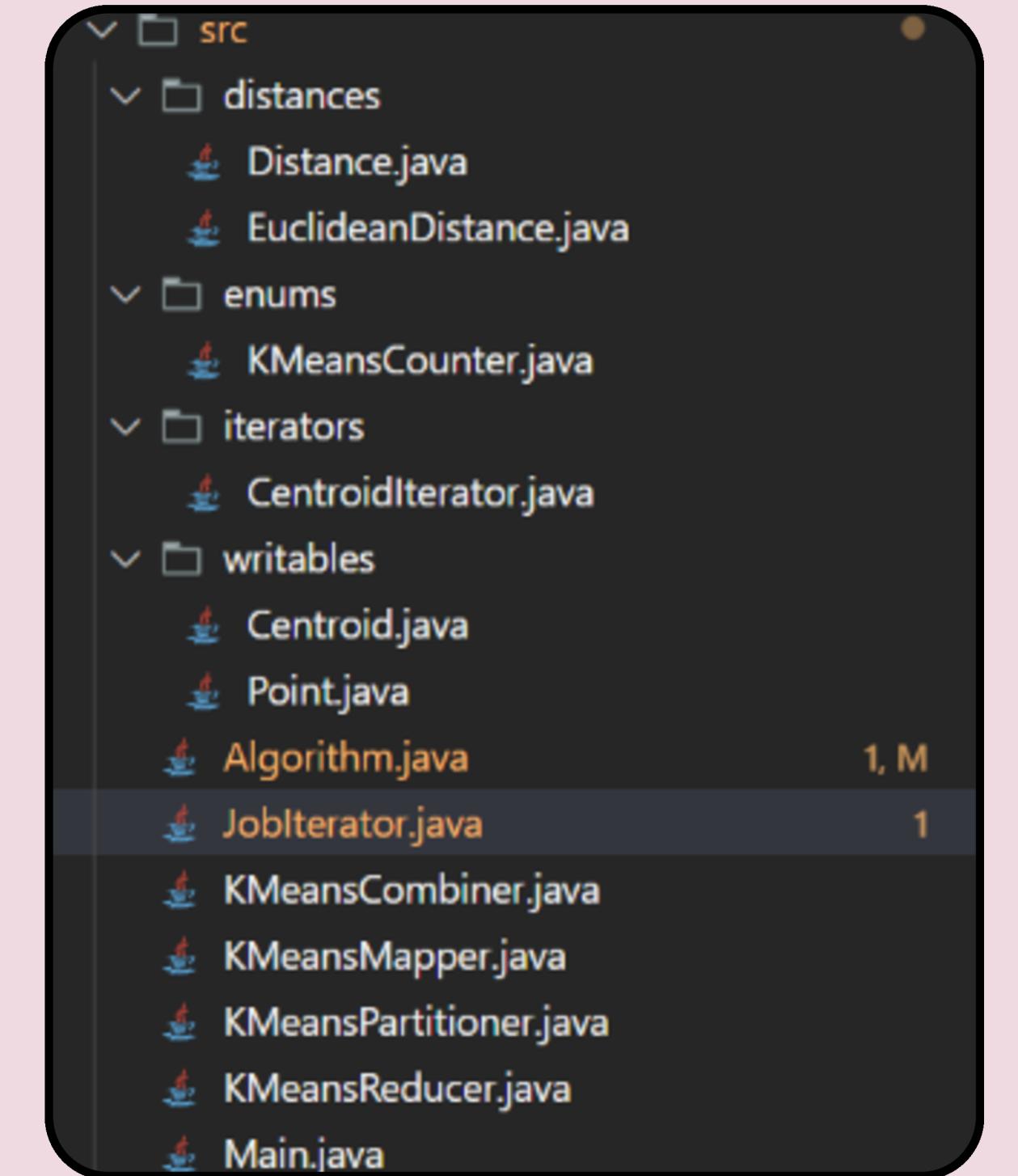
Ứng dụng Map Reduce trong Phân cụm ảnh

3. Triển khai

Thuật toán KMean MapReduce

Cấu trúc thư mục:

- Algorithm.java: Kiểm tra hội tụ hoặc số vòng lặp, khởi tạo và chạy JobIterator.
- JobIterator.java: Xử lý vòng lặp, cập nhật Centroid từ output, xóa output cũ.
- KMeansMapper: Pha Map.
- KMeansCombiner: Pha Combiner.
- KMeansPartitioner: Phân tách Key.
- KMeansReducer: Pha Reduce.
- Distance.java: Tính khoảng cách Centroid - Point.
- CentroidIteration.java: Tìm Points gần nhất với từng Centroid.
- Centroid.java: Lưu thông tin Centroid (tọa độ, nhãn).
- Point.java: Lưu thông tin Point (tọa độ, số lượng Points).



Ứng dụng Map Reduce trong Phân cụm ảnh

3. Triển khai

Chuẩn bị dữ liệu

Mỗi ảnh lưu điểm dữ liệu trong `points.txt`, mỗi điểm ảnh là 3 giá trị màu RGB, cụm khởi tạo ngẫu nhiên.

```
points_path = join(dst_folder_points, f'points_{i}.txt')
clusters_path = join(dst_folder_clusters, f'clusters_{i}.txt')

# load and write points
img = cv2.imread(src_img_path).reshape((-1, 3)).astype(np.float32)
with open(points_path, 'w') as f:
    f.write(nparray_to_str(img))
print(f'Points saved in: {points_path}')
```

Ví dụ về một số point:

226.0	226.0	226.0
228.0	228.0	228.0
230.0	230.0	230.0
230.0	230.0	230.0
231.0	231.0	231.0
232.0	232.0	232.0

Ứng dụng Map Reduce trong Phân cụm ảnh

3. Triển khai

Chuẩn bị dữ liệu

Sinh ngẫu nhiên K cụm, mỗi cụm giống một point và gán nhãn từ số pixel nhỏ nhất đến lớn nhất trong ảnh.

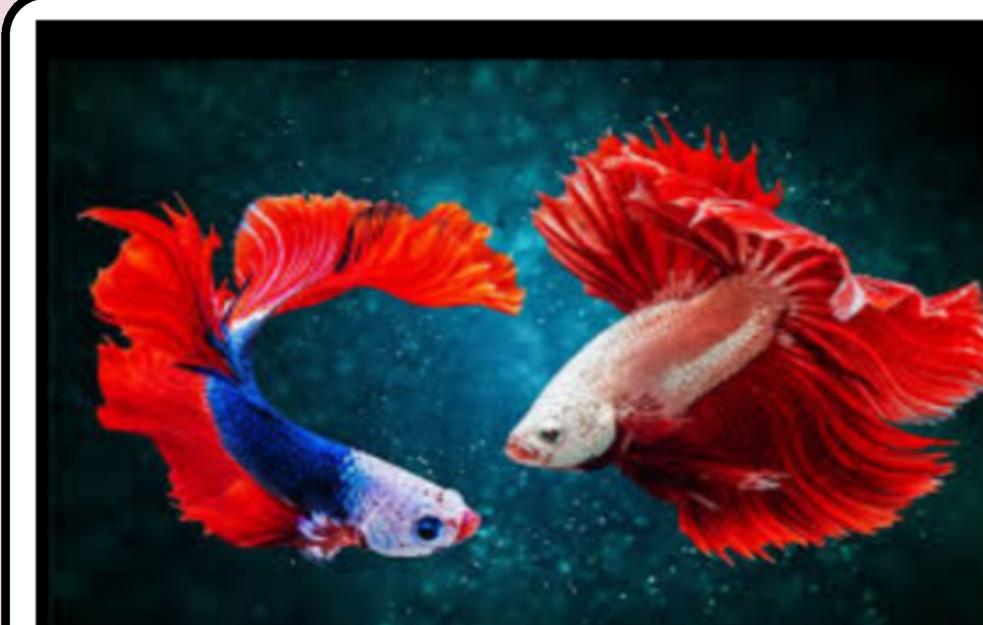
```
s = np.random.uniform(low=img.min(), high=img.max(), size=(k, 3))
tmp_labels = np.arange(1, k + 1).reshape((k, 1))
clusters = np.hstack((tmp_labels, s))

with open(clusters_path, 'w') as f:
    f.write(nparray_to_str(clusters))
print(f'Centroids saved in: {clusters_path}')
```

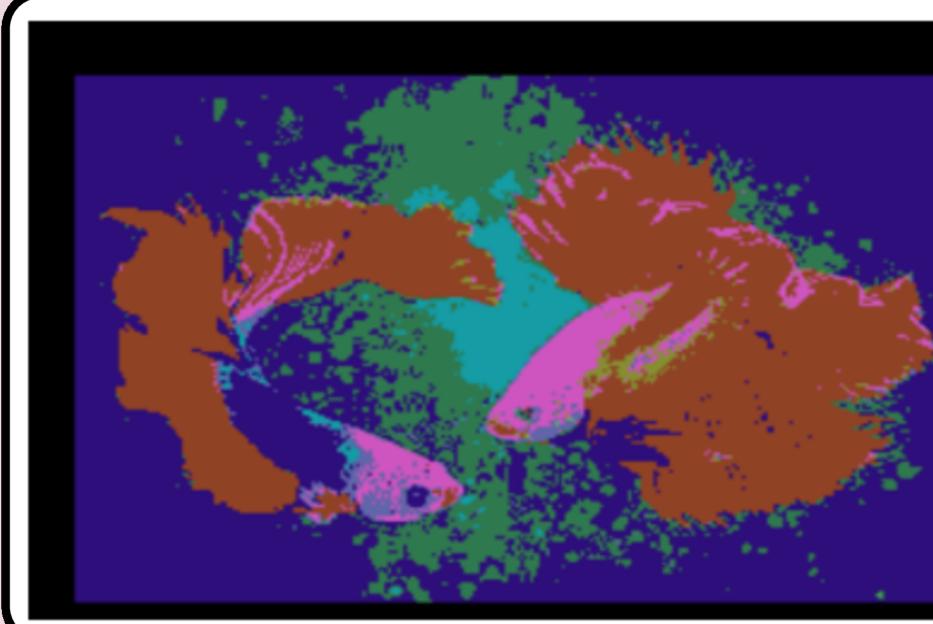
Ứng dụng Map Reduce trong Phân cụm ảnh

4. Thực nghiệm

Hình ảnh ban đầu



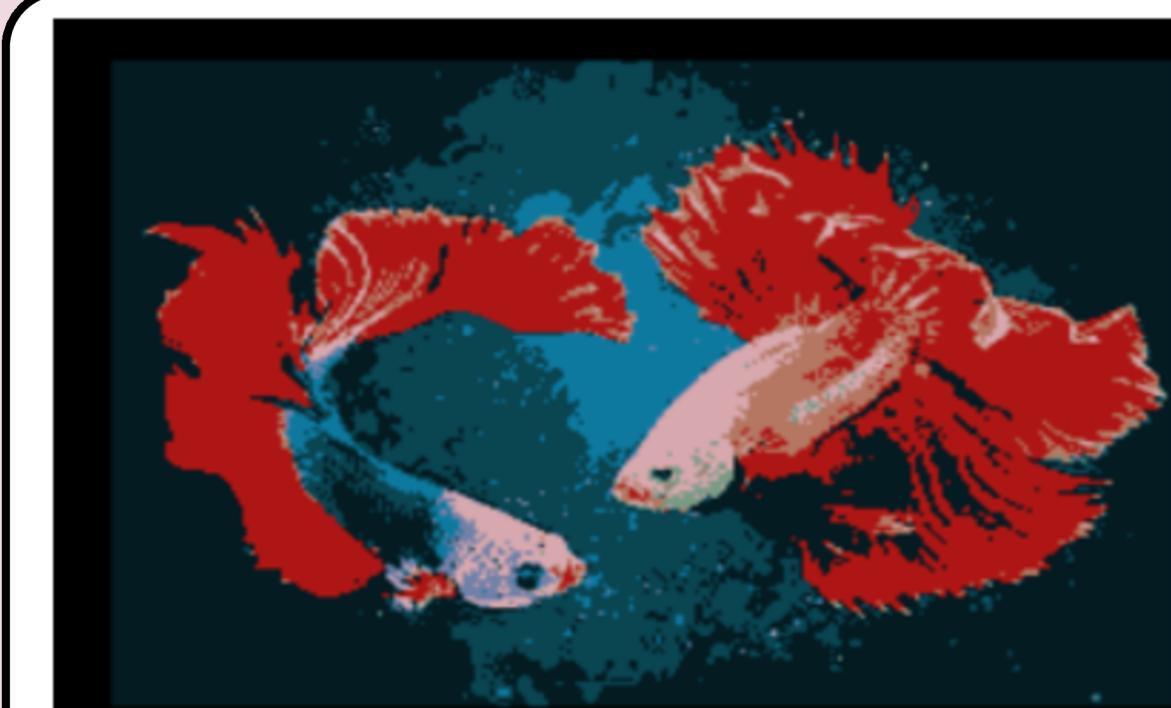
Hình ảnh trước khi chạy MapReduce (10 cụm)



Ứng dụng Map Reduce trong Phân cụm ảnh

4. Thực nghiệm

Hình ảnh sau khi chạy MapReduce với số lần lặp 30





KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

- Ý nghĩa
- Hướng phát triển

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Ý nghĩa

Tổ chức dữ liệu phi cấu trúc

- Biến dữ liệu thành nhóm hợp lý.
- VD: Nhóm “chân dung”, “phong cảnh”.



Hỗ trợ phân tích xu hướng

- Tìm mẫu hình tiềm ẩn.
- VD: Sản phẩm giống nhau.



Cải thiện học sâu

- Tạo dữ liệu có tổ chức cho AI.



Ứng dụng chuyên biệt

- Y tế: MRI, X-quang.
- An ninh: Camera giám sát.



KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

2. Hướng phát triển

- Học sâu (Deep Learning): Kết hợp Autoencoder, CNNs để cải thiện chất lượng phân cụm.
- Học tự giám sát (Self-supervised Learning): Sử dụng Contrastive Learning khi dữ liệu không có nhãn.
- Phân cụm theo ngữ cảnh: Kết hợp thông tin từ văn bản, metadata để cải thiện độ chính xác.
- Ứng dụng thời gian thực: Nghiên cứu thuật toán phân cụm nhanh cho dữ liệu hình ảnh thời gian thực.
- Tăng cường hiệu năng với phân tán: Áp dụng Apache Spark, Hadoop cho dữ liệu lớn.
- Xử lý dữ liệu đa phương thức: Kết hợp ảnh với âm thanh, video để tăng tính liên kết.
- Phân cụm theo ý nghĩa sâu (Semantic Clustering): Sử dụng mô hình ngôn ngữ lớn (như CLIP) để phân cụm theo ý nghĩa thay vì đặc điểm vật lý.

Thank you!

