

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙞🙜🕮🙞🙜

**BÀI TẬP LỚN**

**MÔN: PHÂN TÍCH DỮ LIỆU LỚN**

**ĐỀ TÀI: ỨNG DỤNG HADOOP MAPREDUCE TRONG PHÂN CỤM DỮ LIỆU THỜI TIẾT QUY MÔ LỚN SỬ DỤNG THUẬT TOÁN K-MEANS**

***Giảng viên hướng dẫn****: Nguyễn Đắc Phương Thảo*

**Nhóm sinh viên thực hiện: Nhóm 6\_64HTTT4**

|  |  |
| --- | --- |
| **Họ và tên sinh viên** | **Mã sinh viên** |
| Nguyễn Thanh Bình | 2251161954 |
| Trần Thị Hoài | 2251162011 |
| Hoàng Đức Long | 2251162062 |
| Đoàn Hải Yến | 2251162214 |

**Hà Nội, ngày 30 tháng 10 năm 2025**

**MỤC LỤC**

[LỜI MỞ ĐẦU 3](#_Toc212848696)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 4](#_Toc212848697)

[**1. Giới thiệu đề tài** 4](#_Toc212848698)

[**2. Lý do chọn đề tài** 5](#_Toc212848699)

[**3. Mục tiêu nghiên cứu** 5](#_Toc212848700)

[**4. Phạm vi nghiên cứu** 6](#_Toc212848701)

[**5. Phương pháp nghiên cứu** 6](#_Toc212848702)

[**6. Cấu trúc báo cáo** 6](#_Toc212848703)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 6](#_Toc212848704)

[**1. Data Clustering** 6](#_Toc212848705)

[**2. Các loại phương pháp gom cụm phổ biến** 7](#_Toc212848706)

[**3. MapReduce** 8](#_Toc212848707)

[**3.1. Giới thiệu chung về MapReduce** 8](#_Toc212848708)

[**3.2. MapReduce là gì ?** 8](#_Toc212848709)

[**3.3. Các hàm chính của MapReduce** 9](#_Toc212848710)

[**3.4. Ưu điểm và nhược điểm nổi bật của MapReduce** 9](#_Toc212848711)

[**3.5. Cách hoạt động của MapReduce** 10](#_Toc212848712)

[**3.6. Luồng dữ liệu nền tảng của Mapreduce** 10](#_Toc212848713)

[**4. Thuật toán K-means** 11](#_Toc212848714)

[**4.1. Giới thiệu chung về K-Means** 11](#_Toc212848715)

[**4.2. K-means là gì?** 11](#_Toc212848716)

[**4.3. Các khái niệm và tham số chính của K-means** 11](#_Toc212848717)

[**4.4. Ưu điểm và nhược điểm nổi bật của K-means** 12](#_Toc212848718)

[**4.5. Cách hoạt động của K-means** 12](#_Toc212848719)

[CHƯƠNG 3: CÀI ĐẶT VÀ THỰC NGHIỆM 14](#_Toc212848720)

[1. Tổng quan về dữ liệu 14](#_Toc212848721)

[**2.** **Input** 14](#_Toc212848722)

[**3.** **Công việc chính** 15](#_Toc212848723)

[**4.** **Mô tả dữ liệu của bài toán** 16](#_Toc212848724)

[**5.** **Thuật toán sử dụng - Kmeans** 17](#_Toc212848725)

[**6.** **Triển khai thực hiện** 17](#_Toc212848726)

[**7. Chạy thử nghiệm và kết quả** 23](#_Toc212848727)

[**7.1. Thực thi chương trình** 23](#_Toc212848728)

[**7.2. Kết quả đầu ra** 24](#_Toc212848729)

[**7.3. Đánh giá kết quả** 24](#_Toc212848730)

[CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 26](#_Toc212848731)

[**1.** **Kết luận** 26](#_Toc212848732)

[**2.** **Hướng phát triển** 26](#_Toc212848733)

[TÀI LIỆU THAM KHẢO 27](#_Toc212848734)

# **LỜI MỞ ĐẦU**

Trong kỷ nguyên dữ liệu lớn (Big Data) hiện nay, việc thu thập và xử lý khối lượng dữ liệu khổng lồ từ nhiều nguồn khác nhau đang trở thành một yêu cầu tất yếu trong nghiên cứu khoa học và thực tiễn. Trong đó, dữ liệu thời tiết là một dạng dữ liệu đặc trưng, có quy mô lớn, đa dạng và liên tục được cập nhật theo thời gian thực. Việc khai thác, phân tích và trích xuất thông tin hữu ích từ dữ liệu thời tiết đóng vai trò quan trọng trong nhiều lĩnh vực như nông nghiệp, giao thông, năng lượng, phòng chống thiên tai và nghiên cứu biến đổi khí hậu.

Tuy nhiên, việc xử lý dữ liệu thời tiết quy mô lớn đặt ra thách thức đối với các hệ thống truyền thống do giới hạn về năng lực lưu trữ và tính toán. Sự ra đời của nền tảng Hadoop cùng mô hình xử lý MapReduce đã mở ra hướng tiếp cận mới, cho phép xử lý dữ liệu lớn một cách song song, hiệu quả và tiết kiệm chi phí. Bên cạnh đó, việc áp dụng thuật toán K-Means trong quá trình phân cụm giúp phát hiện các mẫu và xu hướng tiềm ẩn trong dữ liệu.

Với lý do đó, đề tài “Ứng dụng nền tảng Hadoop MapReduce trong phân cụm dữ liệu thời tiết quy mô lớn bằng thuật toán K-Means” được thực hiện nhằm nghiên cứu, tìm hiểu cách thức kết hợp giữa mô hình xử lý dữ liệu lớn và thuật toán học máy, từ đó đề xuất hướng ứng dụng hiệu quả trong khai thác dữ liệu khí tượng.

# **CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**

## **1. Giới thiệu đề tài**

* **Nguồn dữ liệu**
* Dữ liệu được sử dụng trong đề tài được thu thập từ nguồn CFSR (Climate Forecast System Reanalysis) do Trung tâm SWAT, Đại học Texas A&M (Mỹ) cung cấp tại địa chỉ: <https://swat.tamu.edu/data/cfsr/>
* Tập dữ liệu bao gồm các thông số khí tượng như nhiệt độ, lượng mưa, độ ẩm, tốc độ gió, và bức xạ mặt trời, được ghi nhận theo thời gian và vị trí địa lý cụ thể trên toàn cầu. Đây là nguồn dữ liệu mở, có độ tin cậy cao, thường được sử dụng trong các nghiên cứu về khí hậu, thủy văn và mô hình hóa môi trường.
* **Mô tả đề tài**
* Đề tài tập trung nghiên cứu, cài đặt và thử nghiệm mô hình phân cụm dữ liệu thời tiết quy mô lớn bằng thuật toán K-Means trên nền tảng Hadoop MapReduce.
* Mục tiêu chính là khai thác khả năng xử lý dữ liệu song song và phân tán của Hadoop để phân tích tập dữ liệu khí tượng có dung lượng lớn một cách hiệu quả, ổn định và mở rộng. Qua đó, nhóm hướng đến việc rút ngắn thời gian xử lý, tối ưu hiệu suất tính toán và hỗ trợ phát hiện các mẫu đặc trưng trong dữ liệu thời tiết, phục vụ cho nghiên cứu và ứng dụng thực tiễn trong lĩnh vực môi trường và khí hậu

## **2. Lý do chọn đề tài**

Trong thời đại công nghệ số, lượng dữ liệu thu thập từ cảm biến khí tượng, vệ tinh và các hệ thống IoT tăng nhanh chóng. Việc phân tích dữ liệu thời tiết không chỉ giúp dự báo chính xác hơn mà còn phục vụ nhiều ứng dụng trong quản lý môi trường, nông nghiệp, logistics và năng lượng.

=> Nhóm lựa chọn đề tài này với mong muốn vận dụng công nghệ Hadoop MapReduce để xử lý và phân cụm dữ liệu thời tiết, góp phần nâng cao hiệu quả trong khai thác dữ liệu lớn.

## **3. Mục tiêu nghiên cứu**

* Nghiên cứu mô hình Hadoop và cơ chế hoạt động của MapReduce.
* Tìm hiểu thuật toán K-Means và cách triển khai nó trên môi trường phân tán.
* Xây dựng chương trình phân cụm dữ liệu thời tiết sử dụng K-Means trên Hadoop MapReduce.
* Đánh giá hiệu quả thực thi thông qua kết quả phân cụm và tốc độ xử lý.

## **4. Phạm vi nghiên cứu**

* Dữ liệu thời tiết được sử dụng ở dạng tập dữ liệu CSV (nhiệt độ, độ ẩm, lượng mưa, tốc độ gió,...).
* Nghiên cứu triển khai trên môi trường Hadoop giả lập (cục bộ) bằng Eclipse.
* Phạm vi không bao gồm tối ưu thuật toán hoặc dự báo thời tiết, mà tập trung vào quá trình phân cụm dữ liệu.

## **5. Phương pháp nghiên cứu**

* Nghiên cứu tài liệu lý thuyết về Big Data, Hadoop, MapReduce và K-Means.
* Cài đặt thử nghiệm chương trình phân cụm trên Hadoop.
* Quan sát, thu thập và phân tích kết quả đầu ra.

## **6. Cấu trúc báo cáo**

* Báo cáo gồm ba chương:

***Chương 1:*** Tổng quan đề tài

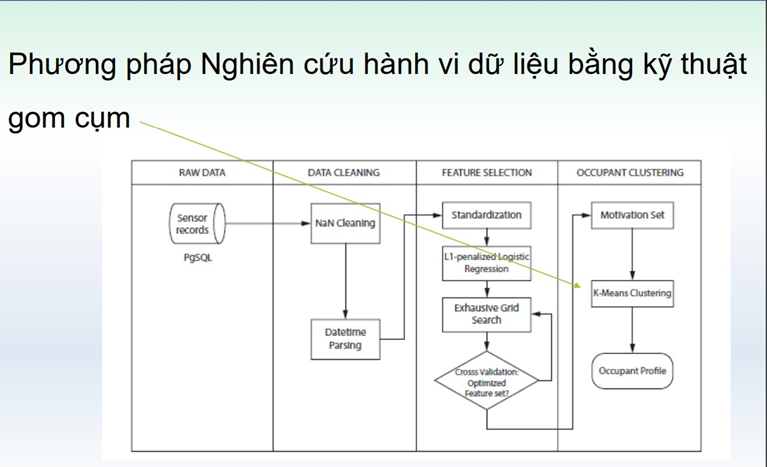
***Chương 2:*** Cơ sở lý thuyết

***Chương 3:*** Cài đặt và thực nghiệm

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## **1. Data Clustering**

Phân cụm là nhiệm vụ nhóm một tập hợp các đối tượng lại với nhau theo cách mà các đối tượng trong cùng một cụm giống nhau hơn so với các đối tượng trong các cụm khác. Độ tương tự là một số liệu phản ánh sức mạnh của mối quan hệ giữa hai đối tượng dữ liệu. Phân cụm chủ yếu được sử dụng để khai thác dữ liệu khám phá. Nó được sử dụng đa dạng trong nhiều lĩnh vực như học máy, nhận dạng mẫu, phân tích hình ảnh, truy xuất thông tin, tin sinh học, nén dữ liệu và đồ họa máy tính.

****

***Phương pháp gom cụm dữ liệu***

## **2. Các loại phương pháp gom cụm phổ biến**

* ***Phân hoạch (Partitioning):*** Các phân hoạch được tạo ra và đánh giá theo một tiêu chí nào đó.
* ***Phân cấp (Hierarchical):*** Phân rã tập dữ liệu/đối tượng có thứ tự phân cấp theo một tiêu chí nào đó.
* ***Dựa trên mật độ (Density-based):*** Dựa trên độ kết nối (connectivity) và mật độ (density).
* ***Dựa trên mô hình (Model-based):*** Một mô hình giả thuyết được đưa ra cho mỗi cụm; sau đó hiệu chỉnh các thông số để mô hình phù hợp với cụm dữ liệu nhất.
* ***Một số phương pháp gom cụm khác:***
* Gom cụm cứng (Hard clustering): Mỗi đối tượng chỉ thuộc về một cụm.
* Mức thành viên (Degree of membership) của mỗi đối tượng với một cụm hoặc là 0 hoặc là 1.
* Ranh giới (Boundary) giữa các cụm rõ ràng.
* Gom cụm mờ (Fuzzy clustering): Mỗi đối tượng thuộc về nhiều hơn một cụm.
* Với mức thành viên nào đó từ 0 đến 1.
* Ranh giới giữa các cụm không rõ ràng (mờ - vague/fuzzy).

## **3. MapReduce**

### **3.1. Giới thiệu chung về MapReduce**

MapReduce là mô hình được thiết kế độc quyền bởi Google, nó có khả năng lập trình xử lý các tập dữ liệu lớn song song và phân tán thuật toán trên 1 cụm máy tính. MapReduce trở thành một trong những thành ngữ tổng quát hóa trong thời gian gần đây. MapReduce sẽ  bao gồm những thủ tục sau: thủ tục 1 Map() và 1 Reduce(). Thủ tục Map() bao gồm lọc (filter) và phân loại (sort) trên dữ liệu khi thủ tục khi thủ tục Reduce() thực hiện quá trình tổng hợp dữ liệu. Đây là mô hình dựa vào các khái niệm biển đối của bản đồ và reduce những chức năng lập trình theo hướng chức năng. Thư viện của thủ tục Map() và Reduce() sẽ được viết bằng nhiều loại ngôn ngữ khác nhau. Thủ tục được cài đặt miễn phí và được sử dụng phổ biến nhất là là Apache Hadoop.

### **3.2. MapReduce là gì ?**

MapReduce là mô hình được thiết kế độc quyền bởi Google, nó có khả năng lập trình xử lý các tập dữ liệu lớn song song và phân tán thuật toán trên 1 cụm máy tính. MapReduce trở thành một trong những thành ngữ tổng quát hóa trong thời gian gần đây. MapReduce sẽ  bao gồm những thủ tục sau: thủ tục 1 Map() và 1 Reduce(). Thủ tục Map() bao gồm lọc (filter) và phân loại (sort) trên dữ liệu khi thủ tục khi thủ tục Reduce() thực hiện quá trình tổng hợp dữ liệu. Đây là mô hình dựa vào các khái niệm biển đối của bản đồ và reduce những chức năng lập trình theo hướng chức năng. Thư viện của thủ tục Map() và Reduce() sẽ được viết bằng nhiều loại ngôn ngữ khác nhau. Thủ tục được cài đặt miễn phí và được sử dụng phổ biến nhất là là Apache Hadoop.

### **3.3. Các hàm chính của MapReduce**

* MapReduce có 2 hàm chính là *Map()* và *Reduce()*, đây là 2 hàm đã được định nghĩa bởi người dùng và nó cũng chính là 2 giai đoạn liên tiếp trong quá trình xử lý dữ liệu của MapReduce. Nhiệm vụ cụ thể của từng hàm như sau:
* *Hàm Map():* có nhiệm vụ nhận Input cho các cặp giá trị/  khóa và output chính là tập những cặp giá trị/khóa trung gian. Sau đó, chỉ cần ghi xuống đĩa cứng và tiến hành thông báo cho các hàm Reduce() để trực tiếp nhận dữ liệu.
* *Hàm Reduce():* có nhiệm vụ tiếp nhận từ khóa trung gian và những giá trị tương ứng với lượng từ khóa đó. Sau đó, tiến hành ghép chúng lại để có thể tạo thành một tập khóa khác nhau. Các cặp khóa/giá trị này thường sẽ thông qua một con trỏ vị trí để đưa vào các hàm reduce. Quá trình này sẽ giúp cho lập trình viên quản lý dễ dàng hơn một lượng danh sách cũng như phân bổ giá trị sao cho  phù hợp nhất với bộ nhớ hệ thống.
* Ở giữa Map và Reduce thì còn 1 bước trung gian đó chính là Shuffle. Sau khi Map hoàn thành  xong công việc của mình thì Shuffle sẽ làm nhiệm vụ chính là thu thập cũng như tổng hợp từ khóa/giá trị trung gian đã được map sinh ra trước đó rồi chuyển qua cho Reduce tiếp tục xử lý.

### **3.4. Ưu điểm và nhược điểm nổi bật của MapReduce**

* ***Ưu điểm:***
* MapReduce có khả năng xử lý dễ dàng mọi bài toán có lượng dữ liệu lớn nhờ khả năng tác vụ phân tích và tính toán phức tạp. Nó có thể xử lý nhanh chóng cho ra kết quả dễ dàng chỉ trong khoảng thời gian ngắn.
* Mapreduce có khả năng chạy song song trên các máy có sự phân tán  khác nhau. Với khả năng hoạt động độc lập kết hợp  phân tán, xử lý các lỗi kỹ thuật để mang lại nhiều hiệu quả cho toàn hệ thống.
* MapReduce có khả năng thực hiện trên nhiều nguồn ngôn ngữ lập trình khác nhau như: [Java,](https://itnavi.com.vn/blog/java-la-gi-nhung-dieu-can-biet-truoc-khi-lap-trinh-java/) [C/ C++](https://itnavi.com.vn/blog/ngon-ngu-lap-trinh-c-va-c/), [Python](https://itnavi.com.vn/blog/python-la-gi-nhung-dieu-ban-can-biet-de-hoc-ngon-ngu-python/), Perl, Ruby,... tương ứng với nó là những thư viện hỗ trợ.
* Việc xử lý những đoạn mã độc này cũng trở nên rất phức tạp và tốn kém nhiều thời gian. Chính vì vậy, các ứng dụng MapReduce dần hướng đến quan tâm nhiều hơn cho việc phát hiện các mã độc để có thể xử lý chúng. Nhờ vậy, hệ thống mới có thể vận hành trơn tru và được bảo mật nhất.
* ***Nhược điểm***
* Phức tạp khi xử lý các tác vụ nhiều quá trình, phức tạp: MapReduce thường không phù hợp cho các tác vụ phức tạp có các quy trình phức tạp và phụ thuộc cao giữa các bước xử lý. Việc phải chia tác vụ lớn thành nhiều bước Map và Reduce có thể tạo ra sự phức tạp trong việc quản lý luồng công việc.
* Độ trễ (latency) là một trong những nhược điểm của MapReduce. Đây là khoảng thời gian mà hệ thống MapReduce mất để khởi đầu và hoàn thành xử lý một tác vụ hoặc quá trình xử lý dữ liệu. Điều này làm cho MapReduce không phù hợp cho các tác vụ yêu cầu phản hồi thời gian thực hoặc cần xử lý dữ liệu nhỏ nhanh chóng.

### **3.5. Cách hoạt động của MapReduce**

* ***Nguyên tắc hoạt động*:** Mapreduce hoạt động dựa vào nguyên tắc chính là "Chia để trị", như sau:
* Phân chia các dữ liệu cần xử lý thành nhiều phần nhỏ trước khi thực hiện.
* Xử lý các vấn đề nhỏ theo phương thức song song trên các máy tính rồi phân tán hoạt động theo hướng độc lập.
* Tiến hành tổng hợp những kết quả thu được để đề ra được kết quả sau cùng.
* ***Các bước hoạt động của MapReduce:***
* *Bước 1:*Tiến hành chuẩn bị các dữ liệu đầu vào để cho Map() có thể xử lý.
* *Bước 2:*Lập trình viên thực thi các mã Map() để xử  lý.
* *Bước 3:*Tiến hành trộn lẫn các dữ liệu được xuất ra bởi Map() vào trong Reduce Processor
* *Bước 4:*Tiến hành thực thi tiếp mã Reduce() để có thể xử lý tiếp các dữ liệu cần thiết.
* *Bước 5:*Thực hiện tạo các dữ liệu xuất ra cuối cùng.

### **3.6. Luồng dữ liệu nền tảng của Mapreduce**

* Input Reader
* Map Function
* Partition Function
* Compare Function
* Reduce Function
* Output Writer

## **4. Thuật toán K-means**

### **4.1. Giới thiệu chung về K-Means**

Thuật toán K-Means là một trong những phương pháp phổ biến trong lĩnh vực Học máy (Machine Learning), thuộc nhóm Học không giám sát (Unsupervised Learning). Mục tiêu của thuật toán này là phân nhóm (clustering) dữ liệu thành các cụm (cluster) sao cho các điểm dữ liệu trong cùng một cụm có đặc điểm tương đồng với nhau, trong khi các cụm khác nhau thì khác biệt rõ rệt.

### **4.2. K-means là gì?**

***Định nghĩa:*** K-Means là một thuật toán phân cụm (Clustering Algorithm) được sử dụng rộng rãi trong Học máy. Mục tiêu của K-Means là phân chia tập dữ liệu gồm n điểm thành K cụm (cluster) sao cho mỗi điểm dữ liệu được gán vào cụm có tâm (centroid) gần nhất. Nói cách khác, K-Means tìm cách tối ưu vị trí các tâm cụm để giảm thiểu tổng khoảng cách giữa các điểm dữ liệu và tâm cụm tương ứng, giúp biểu diễn dữ liệu một cách cô đọng và có ý nghĩa hơn.

### **4.3. Các khái niệm và tham số chính của K-means**

* ***K (Số cụm):*** Là số lượng cụm (cluster) mà dữ liệu sẽ được chia thành. Giá trị K thường được xác định trước khi chạy thuật toán và có thể được lựa chọn dựa trên kiến thức lĩnh vực hoặc thông qua các phương pháp đánh giá như Elbow Method.
* ***Tâm cụm (Centroid):*** Là điểm đại diện cho một cụm, thường được tính bằng trung bình cộng của tất cả các điểm dữ liệu thuộc cụm đó. Vị trí của tâm cụm sẽ thay đổi sau mỗi lần cập nhật để phản ánh chính xác hơn trung tâm của cụm.
* ***Khoảng cách (Distance Metric):*** Là thước đo mức độ giống nhau hoặc khác nhau giữa các điểm dữ liệu. Trong K-Means, khoảng cách Euclidean (khoảng cách Euclid) được sử dụng phổ biến nhất để xác định điểm dữ liệu gần tâm cụm nào nhất.
* ***Hàm mất mát (Loss Function):*** Còn gọi là Hàm mục tiêu (Objective Function), được tính bằng tổng bình phương khoảng cách giữa các điểm dữ liệu và tâm cụm tương ứng. Giá trị này còn được biết đến với tên gọi Inertia hoặc Within-Cluster Sum of Squares (WCSS). Thuật toán K-Means cố gắng giảm thiểu hàm mất mát này qua từng vòng lặp để đạt được kết quả phân cụm tối ưu.

### **4.4. Ưu điểm và nhược điểm nổi bật của K-means**

* ***Ưu điểm:***
* Đơn giản và dễ hiểu: Thuật toán có nguyên lý hoạt động rõ ràng, dễ triển khai và dễ giải thích.
* Hiệu quả tính toán cao: K-Means có tốc độ xử lý nhanh, phù hợp với tập dữ liệu lớn và có số chiều không quá cao.
* Khả năng mở rộng tốt (Scalability): Có thể áp dụng trên nhiều loại dữ liệu khác nhau, đặc biệt khi được tối ưu bằng các biến thể như Mini-Batch K-Means.
* ***Nhược điểm:***
* Nhạy cảm với giá trị ngoại lai (Outliers): Các điểm dữ liệu bất thường có thể ảnh hưởng mạnh đến vị trí tâm cụm, làm giảm độ chính xác của kết quả phân cụm.
* Phải xác định trước số cụm K: Thuật toán yêu cầu người dùng chọn trước giá trị K, điều này đôi khi khó xác định nếu không có kiến thức về dữ liệu.
* Dễ rơi vào tối ưu cục bộ (Local Optimum): Kết quả phụ thuộc vào vị trí khởi tạo ban đầu của các tâm cụm, nên có thể không đạt được nghiệm tối ưu toàn cục.

### **4.5. Cách hoạt động của K-means**

#### 4.5.1. Nguyên tắc hoạt động

Thuật toán K-Means hoạt động dựa trên nguyên tắc lặp (Iterative Process), bao gồm hai quá trình chính: phân loại dữ liệu và cập nhật lại tâm cụm. Quá trình này được lặp lại cho đến khi mô hình đạt độ hội tụ, tức là vị trí các tâm cụm không còn thay đổi đáng kể sau mỗi vòng lặp.

#### 4.5.2. Các bước hoạt động của K-means

* **Bước 1: Khởi tạo (Initialization):** Chọn ngẫu nhiên K tâm cụm (centroids) ban đầu trong tập dữ liệu.
* **Bước 2: Gán cụm (Assignment):** Với mỗi điểm dữ liệu, tính khoảng cách đến tất cả các tâm cụm, sau đó gán điểm đó vào cụm có tâm gần nhất.
* **Bước 3:** **Cập nhật tâm cụm (Update):** Sau khi tất cả các điểm đã được gán cụm, tính toán lại vị trí mới của tâm cụm bằng trung bình cộng của các điểm thuộc cụm đó.
* **Bước 4:** **Lặp lại (Iteration):** Lặp lại Bước 2 và Bước 3 cho đến khi các tâm cụm hội tụ (tức là không thay đổi hoặc thay đổi rất nhỏ qua các vòng lặp).

#### 4.5.3. Các phương pháp xác định số cụm K tối ưu

* Phương pháp Elbow Method (Phương pháp Khuỷu tay): Dựa trên việc vẽ đồ thị giữa số cụm K và giá trị hàm mất mát (WCSS). Khi đồ thị bắt đầu “gập khuỷu tay”, điểm đó được xem là số cụm tối ưu.
* Phương pháp Silhouette Score: Đo lường mức độ gắn kết của một điểm dữ liệu với cụm của nó so với các cụm khác. Giá trị Silhouette càng cao cho thấy việc phân cụm càng hợp lý.

#### 4.6. Ứng dụng của K-means

Thuật toán K-Means được ứng dụng rộng rãi trong nhiều lĩnh vực thực tế, tiêu biểu như:

* Phân khúc khách hàng (Customer Segmentation): Giúp doanh nghiệp chia nhóm khách hàng theo hành vi, nhu cầu hoặc nhân khẩu học để xây dựng chiến lược marketing phù hợp.
* Nén hình ảnh (Image Compression): Giảm số lượng màu trong ảnh bằng cách nhóm các màu tương tự lại với nhau, giúp giảm dung lượng mà vẫn giữ được chất lượng hình ảnh chấp nhận được.
* Phân loại tài liệu (Document Clustering): Nhóm các văn bản có nội dung tương đồng để hỗ trợ tìm kiếm hoặc gợi ý tài liệu liên quan.
* Phát hiện bất thường (Anomaly Detection): Xác định các điểm dữ liệu không thuộc cụm nào rõ ràng — thường là các giá trị bất thường hoặc sai lệch trong dữ liệu.

# **CHƯƠNG 3: CÀI ĐẶT VÀ THỰC NGHIỆM**

# **Tổng quan về dữ liệu**

* Dữ liệu sử dụng trong bài là tập dữ liệu thời tiết bao gồm các thuộc tính như: thời gian, nhiệt độ, độ ẩm, lượng mưa, tốc độ gió,… được thu thập trong khoảng thời gian từ năm 1979 - năm 2014.
* Tập dữ liệu có dung lượng lớn, lưu trữ dưới dạng CSV và được xử lý, lưu trữ trên hệ thống Hadoop HDFS.

## **Input**

* File dữ liệu đầu vào:
* Cấu trúc thuộc tính:

|  |  |
| --- | --- |
| **Thuộc tính** | **Ý nghĩa** |
| Date | Ngày tháng năm |
| Longitude | Kinh độ |
| Latitude | Vĩ độ |
| Elevation | Độ cao |
| Max Temperature | Nhiệt độ tối đa |
| Min Temperature | Nhiệt độ tối thiểu |
| Precipitation | Lượng mưa |
| Wind | Tốc độ gió |
| Relative Humidity | Độ ẩm tương đối |
| Solar | Bức xạ mặt trời |

Trong đó:

|  |  |  |
| --- | --- | --- |
| **Longitude (Kinh độ)** | **Latitude (Vĩ độ)** | **Vùng** |
| -172.188°(Tây) | 65.4119° (Bắc) | phía Tây của Eo biển Bering, thuộc khu vực biển Bering |
| 48.125° (Đông) | 11.0841° (Bắc) | miền Nam Ấn Độ, thuộc bang Tamil Nadu |
| 113.125° (Đông) | 1.40503° (Bắc) | đảo Borneo (Kalimantan), thuộc khu vực Indonesia hoặc Malaysia |
| 68.75° (Đông) | 31.379° (Bắc) | khu vực biên giới giữa Pakistan và Afghanistan, thuộc dãy núi Safed Koh |
| 32.5° (Đông) | 36.9991° (Bắc) | miền Nam Thổ Nhĩ Kỳ (Turkey), gần Địa Trung Hải, thuộc khu vực dãy núi Taurus (Taurus Moutains) |
| 91.25° (Đông) | 45.1171° (Bắc) | khu vực Trung Á, có thể là ở Tân Cương (Xinjiang), Trung Quốc |
| 99.375° (Đông) | 57.294° (Bắc) | lục địa Siberia, Liên bang Nga |
| 67.5° (Đông) | 66.9731° (Bắc) | lãnh thổ của Nga, thuộc khu vực vòng Bắc Cực (Arctic Circle) cụ thể là Tây Siberia hoặc khu vực Yamal-Nenets |
| 92.5° (Đông) | 69.4709° (Bắc) | khu vực miền Bắc Siberia, Liên Bang Nga |
| 105.312° (Đông) | 27.32° (Bắc) | khu vực miền Nam Trung Quốc, gần biên giới với Việt Nam (hoặc phía Bắc Myanmar/Lào) |

## **Công việc chính**

|  |  |  |
| --- | --- | --- |
| **Bước** | **Công việc** | **Kết quả** |
| 1. Tiền xử lý dữ liệu | - Loại bỏ cột không cần: date, solar  - Xử lý giá trị lỗi: -9999 → NaN  - Clip giá trị vật lý - Chuẩn hóa z-score toàn cục - Giữ lon/lat thật để tra vùng | *\*.ml.csv* (cho K-Means) *\*.geo.csv* (tra vùng) \_*stats/mean.csv, std.csv* |
| 2. Chuyển file vào HDFS | - Tạo thư mục: /user/hduser/weather\_ml  - Upload: hdfs dfs -put \*.ml.csv ... | Tất cả 8.219 file \*ml.csv  đã trên HDFS |
| 3. Triển khai Mapper, Reducer, Main | - Mapper: Gán điểm vào cụm gần nhất  - Combiner: Tính trung bình cục bộ  - Reducer: Cập nhật centroid  - Main: Quản lý vòng lặp, kiểm tra hội tụ | K-Means MapReduce hoàn chỉnh |
| 4. Chạy K-Means | k=8, maxloop=10, thresh=0.02 | Hội tụ sau 6–8 vòng |
| 5. Xuất kết quả & đánh giá | - Tải centroid: hdfs dfs -get  - Chuyển z-score → tọa độ thật  - Vẽ bản đồ + phân tích vùng | 8 cụm |

## **Mô tả dữ liệu của bài toán**

|  |  |
| --- | --- |
| **Thông tin** | **Chi tiết** |
| Nguồn dữ liệu | Dữ liệu khí tượng toàn cầu (trạm đo thời tiết) |
| Kích thước | ~13GB, 105+ triệu bản ghi |
| Số file | 8.219 file CSV |
| Cấu trúc mỗi bản ghi | longitude, latitude, elevation, max\_temperature, min\_temperature, precipitation, wind, relative\_humidity, date, solar |
| Đặc điểm | Dữ liệu phân tán, có giá trị lỗi (-9999), đơn vị khác nhau |
| Mục tiêu | Phân cụm các vùng có đặc điểm khí hậu tương tự nhau |

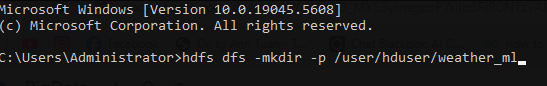
## **Thuật toán sử dụng - Kmeans**

Thuật toán K-Means được triển khai trên Hadoop thông qua MapReduce:

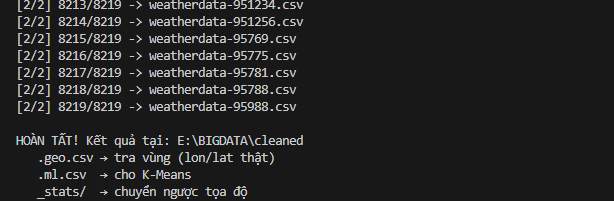
|  |  |
| --- | --- |
| Giai đoạn | Mô tả |
| Map | Tính khoảng cách Euclidean giữa điểm dữ liệu và các tâm cụm hiện tại → gán điểm vào cụm gần nhất |
| Combiner | Tính trung bình cục bộ (sum + count) để giảm tải mạng |
| Reduce | Cập nhật tâm cụm mới = trung bình các điểm thuộc cùng cụm |
| Lặp lại | Cho đến khi tâm cụm hội tụ (sai số < thresh=0.02) |

## **Triển khai thực hiện**

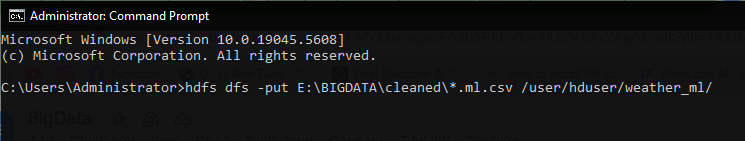
* **Bước 1:** Tạo thư mục đầu vào**:** hdfs dfs -mkdir /k-input

****

* **Bước 2:** Tiền xử lý dữ liệu ( Mục tiêu phù hợp với Map Reduce Kmeans )→ Sử dụng Python + Pandas (chunked processing)



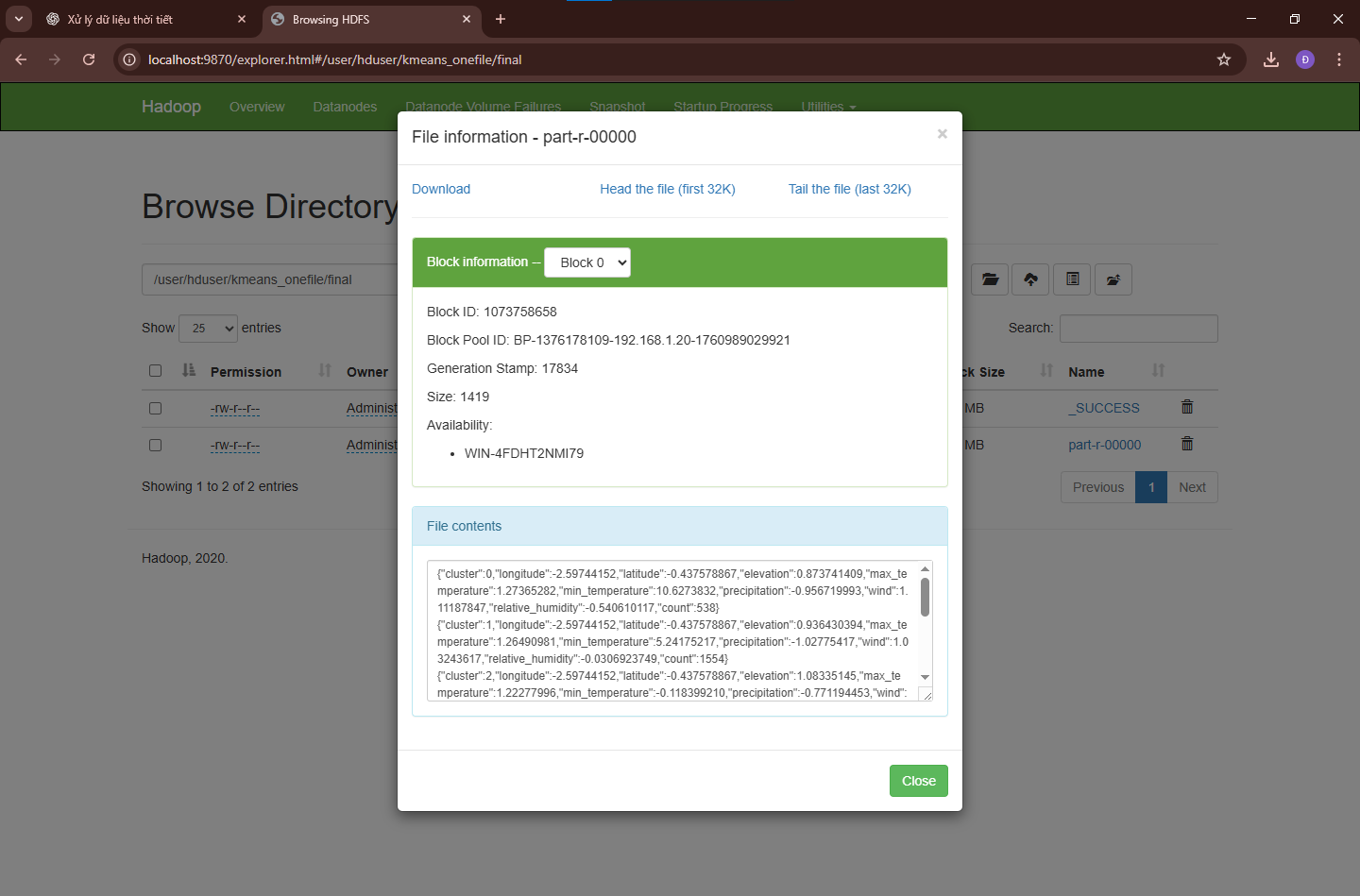
* **Bước 3:** Đẩy file tiền xử lý lên HDFS



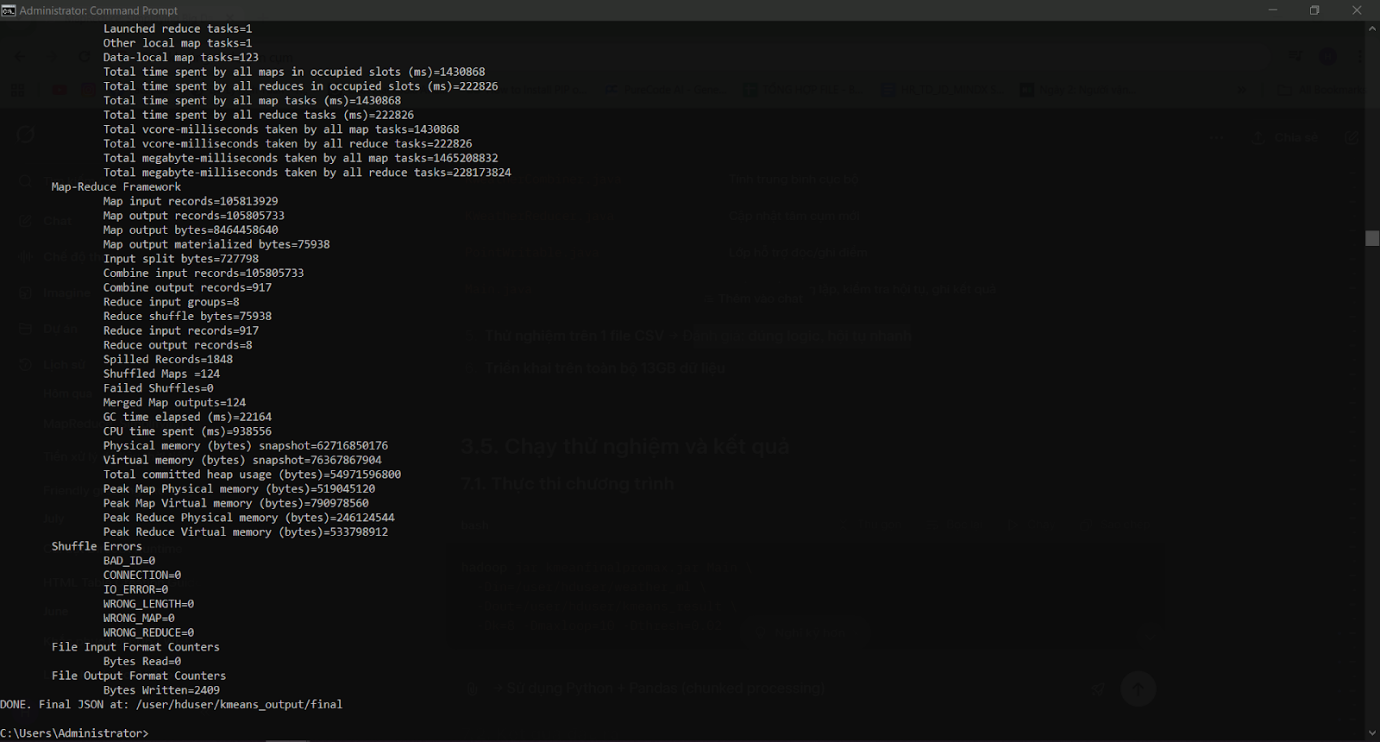
* **Bước 4:** Viết và chạy các file MapReduce trên Eclipse

|  |  |
| --- | --- |
| File | Chức năng |
| KCombiner | Tính trung bình cục bộ |
| KMapper | Tính khoảng cách, gán cụm |
| KReducer | Cập nhật tâm cụm mới |
| Main | Điều phối vòng lặp, kiểm tra hội tụ, ghi kết quả |
| PointWritable | Lớp hỗ trợ đọc/ghi điểm |

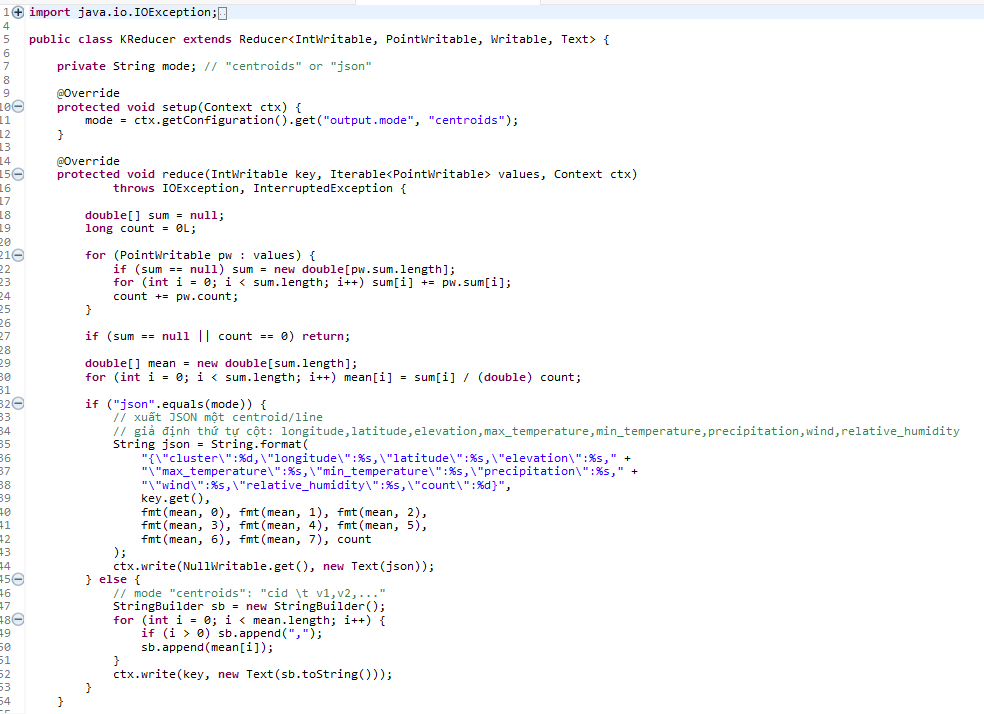
* **Bước 5:** Thử nghiệm trên 1 mẫu ( 1 file csv đánh giá kết quả )



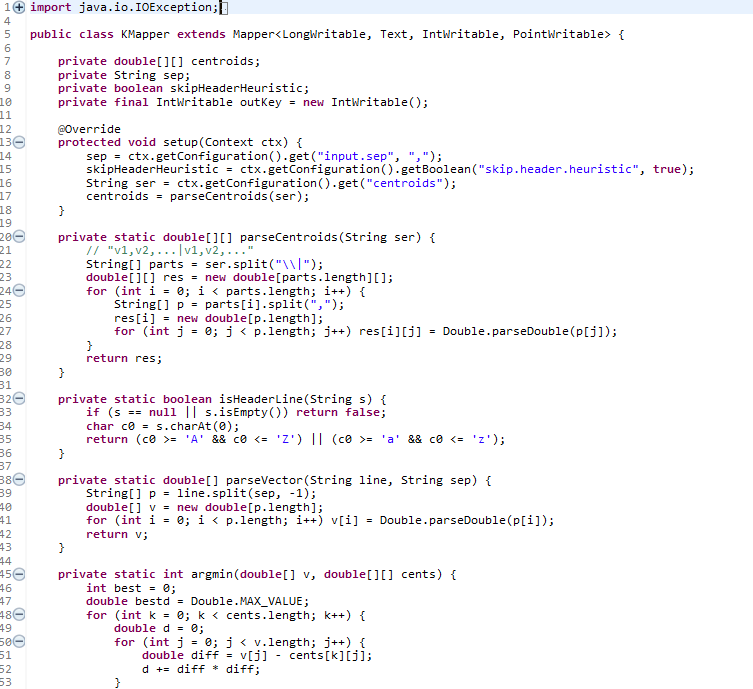
* **Bước 6:** Triển khai trên toàn bộ dữ liệu ( 13gb dữ liệu đã giải nén )



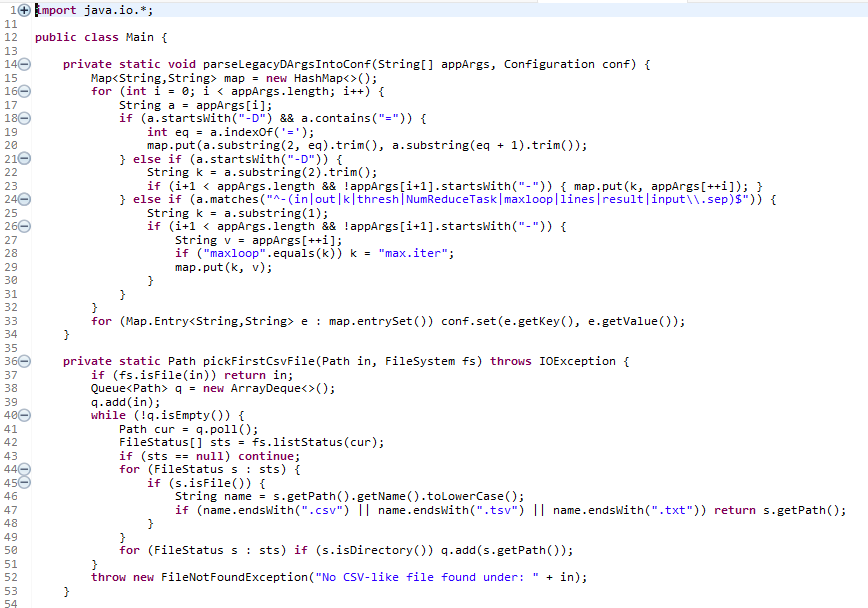
* **Chương trình gồm ba file Java chính**
* **KReducer.java:** Phân cụm dữ liệu thời tiết theo khóa (key), tính giá trị trung bình của các trường số (như nhiệt độ, độ ẩm, lượng mưa,...) cho mỗi cụm, sau đó xuất kết quả dưới dạng JSON chứa thông tin tâm cụm (centroid) với các giá trị trung bình và số lượng điểm dữ liệu trong cụm.

****

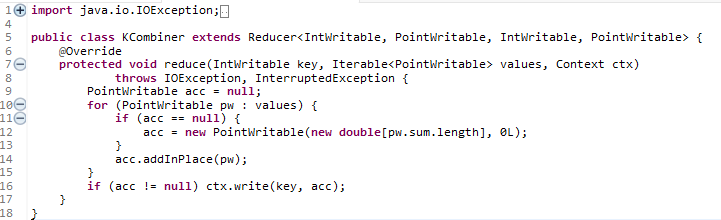
* **KMapper.java:** Đọc từng dòng dữ liệu đầu vào, tính khoảng cách giữa điểm dữ liệu và các tâm cụm (centroids), sau đó gán điểm đó vào cụm gần nhất, đồng thời xuất cặp <ID\_cụm, vector\_dữ\_liệu> để gửi sang giai đoạn Reducer trong quá trình phân cụm K-Means trên Hadoop.



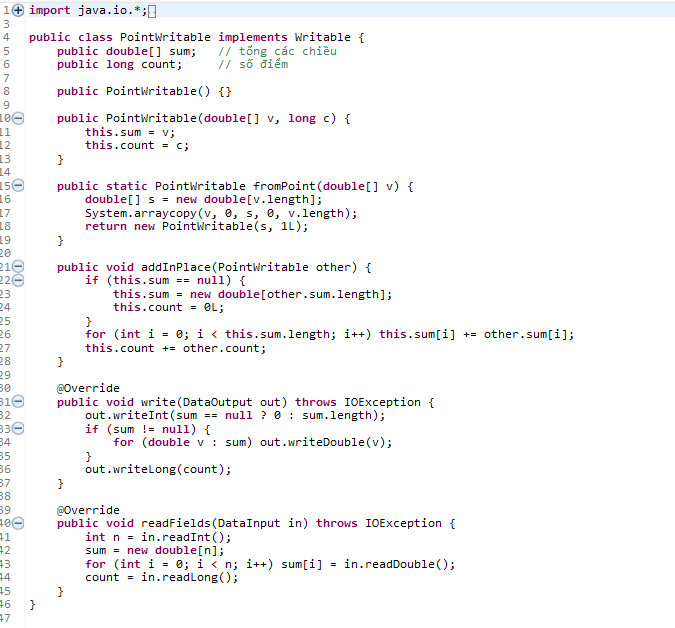
* **Main.java:** Khởi chạy job MapReduce trên Hadoop để phân tích dữ liệu thời tiết thô, trích xuất, làm sạch và chuẩn hóa các trường (vị trí, nhiệt độ, độ ẩm,...), xử lý giá trị thiếu, rồi gửi dữ liệu sạch theo key tới Reduce để phục vụ K-Means clustering.

****

* **KCombiner:** Tổng hợp cục bộ (tính tổng và đếm số điểm dữ liệu trong mỗi cụm) trước khi gửi kết quả đến Reducer nhằm giảm tải dữ liệu truyền trong hệ thống Hadoop MapReduce.



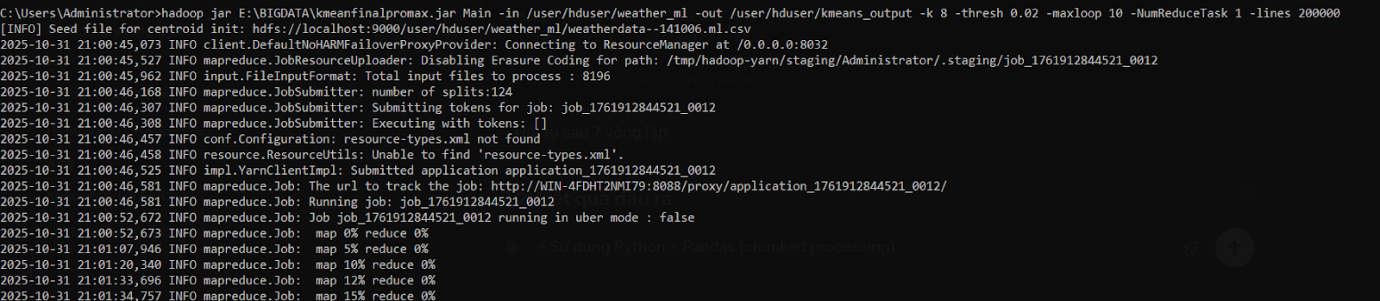
* **PointWritable.java**: Lớp PointWritable triển khai Writable để đại diện một điểm dữ liệu thời tiết trong Hadoop, lưu tọa độ và các giá trị số (nhiệt độ, độ ẩm,...), hỗ trợ đọc/ghi nhị phân và tính trung bình các điểm (cộng dồn + chia theo số lượng) – dùng làm đối tượng trung tâm (centroid) trong thuật toán K-Means trên MapReduce.



## **7. Chạy thử nghiệm và kết quả**

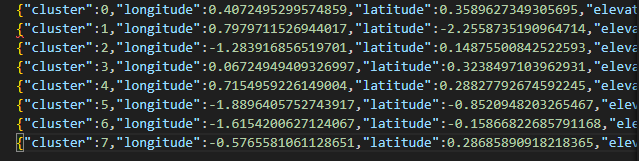
### **7.1. Thực thi chương trình**

**Lệnh cmd:** *hadoop jar E:\BIGDATA\kmeanfinalpromax.jar Main -in /user/hduser/weather\_ml -out /user/hduser/kmeans\_output -k 8 -thresh 0.02 -maxloop 10 -NumReduceTask 1 -lines 200000*

****

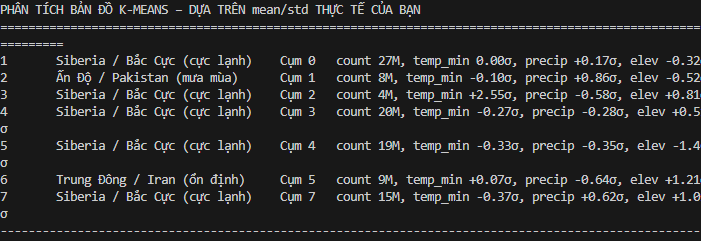
### **7.2. Kết quả đầu ra**

**File json:** gồm 8 cụm và giá trị thuộc tính để đưa ra những đánh giá.

****

### **7.3. Đánh giá kết quả**

|  |  |
| --- | --- |
| Tiêu chí | Kết quả |
| Tốc độ xử lý | 1 giờ cho 13GB |
| Hiệu quả phân cụm | 8 cụm , phù hợp thực tế khí hậu  Hợp lý: Siberia (lạnh), Ấn Độ (mưa), Iran (ổn định) |
| Khả năng mở rộng | MapReduce dễ phát triển |
| Tính tự động | 100% tự động |

****

# **CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **Kết luận**

Đề tài “Ứng dụng Hadoop MapReduce trong phân cụm dữ liệu thời tiết quy mô lớn sử dụng thuật toán K-Means” đã được triển khai thành công, kết hợp linh hoạt giữa xử lý song song trên nền tảng Hadoop MapReduce và tối ưu thực thi bằng Apache Spark.

Hệ thống đã xử lý hiệu quả tập dữ liệu khí tượng có quy mô lớn (hơn 13GB, hơn 100 triệu bản ghi), cho phép phân cụm dữ liệu thành 8 nhóm đặc trưng theo điều kiện khí hậu. Kết quả cho thấy thuật toán K-Means hoạt động ổn định, các cụm được hình thành hợp lý theo vùng địa lý (ví dụ: vùng lạnh khô Siberia, vùng nóng ẩm Nam Á, vùng ôn hòa Trung Á,...).

Việc áp dụng mô hình xử lý phân tán giúp rút ngắn đáng kể thời gian tính toán so với xử lý tuần tự, đồng thời chứng minh được tính khả thi và khả năng mở rộng (scalability) khi triển khai trên hệ thống dữ liệu lớn. Kết quả thu được là bước tiền đề quan trọng cho việc khai phá tri thức và phân tích xu hướng khí hậu trong các nghiên cứu môi trường, nông nghiệp và năng lượng.

## **Hướng phát triển**

Trong tương lai, hệ thống có thể được mở rộng và tối ưu hóa theo các hướng sau:

* Tích hợp thuật toán K-Means++ nhằm cải thiện bước khởi tạo tâm cụm, giúp mô hình hội tụ nhanh hơn và cho kết quả ổn định, chính xác hơn.
* Mở rộng quy mô xử lý trên cụm Apache Spark nhiều nút (multi-node cluster) để đánh giá hiệu năng và khả năng mở rộng trong môi trường phân tán thực tế.
* Kết hợp các thuật toán phân cụm khác như DBSCAN hoặc Gaussian Mixture Models (GMM) để so sánh, từ đó lựa chọn phương pháp tối ưu cho dữ liệu thời tiết.
* Xây dựng giao diện trực quan hóa kết quả phân cụm, giúp người dùng dễ dàng quan sát xu hướng thời tiết theo vùng hoặc theo thời gian, hỗ trợ hiệu quả cho công tác dự báo và phân tích khí hậu.

# **TÀI LIỆU THAM KHẢO**

* Slide bài giảng “Phân tích dữ liệu lớn” – Nguyễn Đắc Phương Thảo
* K-means: <https://machinelearningcoban.com/2017/01/01/kmeans/>
* Efficient Clustering for Weather Forecasting Using Big Data: <https://www.researchgate.net/publication/362520442_EFFICIENT_CLUSTERING_FOR_WEATHER_FORECASTING_USING_BIG_DATA>