

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT VĨNH LONG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO XỬ LÝ DỮ LIỆU LỚN**

**Mã học phần: TH1347**

**Đề tài: Tìm hiểu về Apache Flink và ứng dụng  
trong bài toán xử lý dữ liệu theo luồng**

**Giảng viên hướng dẫn: ThS. Trần Phan An Trường**

**Sinh viên thực hiện:**

- Văn Thị Mỹ Trang                      MSSV: 20004223
- Nguyễn Huỳnh Phụng Thiên      MSSV: 20004197

**Lớp: ĐH.CÔNG NGHỆ THÔNG TIN 2020**

**Khóa: 2020-2024**

Vĩnh Long, năm 2023

## NHẬN XÉT & ĐÁNH GIÁ ĐIỂM CỦA NGƯỜI HƯỚNG DẪN

Ý thức thực hiện: .....

.....  
.....  
.....

Nội dung thực hiện: .....

.....  
.....  
.....

Hình thức trình bày:.....

.....  
.....  
.....

Tổng hợp kết quả:.....

.....  
.....  
.....

☐ Tổ chức báo cáo trước hội đồng

☐ Tổ chức chấm thuyết minh

Vĩnh Long, ngày.....tháng.....năm.....

Người hướng dẫn

(Ký và ghi rõ họ tên)

## **LỜI CẢM ƠN**

Lời đầu tiên, em xin bày tỏ sự kính trọng và lòng biết ơn sâu sắc nhất đến thầy Trần Phan An Trường, giảng viên khoa Công Nghệ Thông Tin Trường Đại học Sư Phạm Kỹ Thuật Vĩnh Long, thầy là người đã trực tiếp hướng dẫn và giúp đỡ để em có thể hoàn thành báo cáo này.

Tuy nhóm chúng em còn nhiều thiếu sót trong quá trình báo cáo, mong thầy và các bạn tận tình đóng góp ý kiến, để nhóm chúng em cải thiện bài báo cáo của nhóm.

Nhóm chúng em sẽ cố gắng hoàn thiện bài luận tốt hơn, chăm chỉ trong quá trình xây dựng bài học, rút kinh nghiệm cho những đề án, bài luận sau này.

Kính chúc thầy nhiều sức khỏe, hạnh phúc thành công trên con đường sự nghiệp giảng dạy.

Chúng em xin chân thành cảm ơn!

Em xin chân thành cảm ơn!

# MỤC LỤC

LỜI CẢM ƠN.....	
MỤC LỤC .....	
LỜI NÓI ĐẦU.....	
CHƯƠNG 1: TỔNG QUAN VỀ DỮ LIỆU LỚN .....	1
1.1. Khái niệm Dữ liệu lớn (Big Data).....	1
1.2. Lịch sử hình thành và phát triển.....	3
1.3. Công nghệ và công cụ xử lý dữ liệu lớn: .....	4
1.4. Tầm quan trọng và ảnh hưởng của dữ liệu lớn: .....	6
1.5. Lợi ích và thách thức của xử lý dữ liệu lớn: .....	6
CHƯƠNG 2: GIỚI THIỆU ĐỀ TÀI.....	8
2.1. Giới thiệu chung về Apache Flink .....	8
2.1.1. Khái niệm Apache Flink .....	8
2.1.2. Lịch sử hình thành và phát triển.....	9
2.2. Kiến trúc của Apache Flink.....	9
2.2.1. Kiến trúc .....	9
2.2.2. Nguyên lý hoạt động .....	12
2.3 Tầm quan trọng của Apache Flink .....	13
2.4 Tính năng.....	13
2.5. Lợi ích .....	14
CHƯƠNG 3: CÀI ĐẶT VÀ THAO TÁC APACHE FLINK.....	16
3.1. Cài đặt Apache Flink trên Ubuntu 20.04 LTS .....	16
3.2. Thao tác Apache Flink trên Ubuntu 20.04 LTS .....	18
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	21

4.1 Kết luận .....	21
4.2 Hướng phát triển.....	21
TÀI LIỆU THAM KHẢO .....	22

## LỜI NÓI ĐẦU

Xử lý dữ liệu lớn đã trở thành một lĩnh vực quan trọng và hấp dẫn trong lĩnh vực khoa học dữ liệu và công nghệ thông tin. Với sự gia tăng nhanh chóng của dữ liệu từ các nguồn như mạng xã hội, thiết bị di động, cảm biến và hệ thống thông tin kinh doanh, việc hiểu và tận dụng dữ liệu này trở thành một thách thức quan trọng đối với các tổ chức và doanh nghiệp.

Qua quá trình học tập và tìm hiểu về xử lý dữ liệu lớn, chúng em thực hiện đề tài “Tìm Hiểu Về Apache Flink Và Ứng Dụng Trong Bài Toán Xử Lý Dữ Liệu Theo Luồng”,

Báo cáo này tập trung vào xử lý dữ liệu lớn và các phương pháp, công nghệ liên quan để hiểu và phân tích dữ liệu khối lượng lớn. Chúng em sẽ xem xét các khái niệm cơ bản, công cụ và kỹ thuật xử lý dữ liệu lớn, cũng như những lợi ích và thách thức của việc làm việc với dữ liệu lớn.

Nhóm em xin chân thành cảm ơn thầy Trần Phan An Trường đã hỗ trợ giúp nhóm em thực hiện đề tài này.

Vì lý do thời gian, kinh nghiệm cũng như trình độ còn hạn chế nên bài báo cáo còn nhiều thiếu sót, nên mong thầy cô và các bạn thông cảm.



+ **Tốc độ (Velocity)** Tốc độ có thể hiểu theo 2 khía cạnh: (a) Khối lượng dữ liệu gia tăng rất nhanh (mỗi giây có tới 72.9 triệu các yêu cầu truy cập tìm kiếm trên web bán hàng của Amazon); (b) Xử lý dữ liệu nhanh ở mức thời gian thực (real-time), có nghĩa dữ liệu được xử lý ngay tức thời ngay sau khi chúng phát sinh (tính đến bằng mili giây). Các ứng dụng phổ biến trên lĩnh vực Internet, Tài chính, Ngân hàng, Hàng không, Quân sự, Y tế – Sức khỏe như hiện nay phần lớn dữ liệu lớn được xử lý real-time. Công nghệ xử lý dữ liệu



lớn ngày nay đã cho phép chúng ta xử lý tức thì trước khi chúng được lưu trữ vào cơ sở dữ liệu.

+ Đa dạng (Variety) Đối với dữ liệu truyền thống chúng ta hay nói đến dữ liệu có cấu trúc, thì ngày nay hơn 80% dữ liệu được sinh ra là phi cấu trúc (tài liệu, blog, hình ảnh, video, bài hát, dữ liệu từ thiết bị cảm biến vật lý, thiết bị chăm sóc sức khỏe...). Big data cho phép liên kết và phân tích nhiều dạng dữ liệu khác nhau. Ví dụ, với các bình luận của một nhóm người dùng nào đó trên Facebook với thông tin video được chia sẻ từ Youtube và Twitter

+ Độ tin cậy/chính xác (Veracity) Một trong những tính chất phức tạp nhất của Dữ liệu lớn là độ tin cậy/chính xác của dữ liệu. Với xu hướng phương tiện truyền thông xã hội (Social Media) và mạng xã hội (Social Network) ngày nay và sự gia tăng mạnh mẽ tính tương tác và chia sẻ của người dùng Mobile làm cho bức tranh xác định về độ tin cậy & chính xác của dữ liệu ngày một khó khăn hơn. Bài toán phân tích và loại bỏ dữ liệu thiếu chính xác và nhiễu đang là tính chất quan trọng của Big data.

+ Giá trị (Value) Giá trị là đặc điểm quan trọng nhất của dữ liệu lớn, vì khi bắt đầu triển khai xây dựng dữ liệu lớn thì việc đầu tiên chúng ta cần phải làm đó là xác định được giá trị của thông tin mang lại như thế nào, khi đó chúng ta mới có quyết định có nên triển khai dữ liệu lớn hay không. Nếu chúng ta có dữ liệu lớn mà chỉ nhận được 1% lợi ích từ nó, thì không nên đầu tư phát triển dữ liệu lớn. Kết quả dự báo chính xác thể hiện rõ nét nhất về giá trị của dữ liệu lớn mang lại. Ví dụ, từ khối dữ liệu phát sinh trong quá trình khám, chữa bệnh sẽ giúp dự báo về sức khỏe được chính xác hơn, sẽ giảm được chi phí điều trị và các chi phí liên quan đến y tế.



Hình 1.2 Đặc trưng của Big Data

## 1.2. Lịch sử hình thành và phát triển

Lịch sử hình thành và phát triển của Big Data có thể được chia thành 3 giai đoạn chính:

- Giai đoạn đầu (1960-2000)

Giai đoạn này đánh dấu sự ra đời của các khái niệm và công nghệ cơ bản cho Big Data, bao gồm:

- ❖ 1960: Cùng với sự phát triển của máy tính, các nhà khoa học bắt đầu nghiên cứu về cách thu thập, lưu trữ và xử lý dữ liệu lớn.
- ❖ 1970: Các thuật toán phân tích dữ liệu thống kê được phát triển để xử lý các tập dữ liệu lớn.
- ❖ 1980: Các hệ thống quản trị cơ sở dữ liệu (DBMS) được phát triển để lưu trữ và truy cập dữ liệu lớn.
- ❖ 1990: Internet và các công nghệ web ra đời, tạo ra nguồn dữ liệu lớn mới.

- Giai đoạn phát triển (2000-2010)

Giai đoạn này đánh dấu sự bùng nổ của Big Data, với sự ra đời của các công nghệ và phương pháp xử lý dữ liệu lớn tiên tiến, bao gồm:

- ❖ 2000: Khái niệm "Big Data" lần đầu tiên được đề cập trong một bài báo của Doug Laney trên tạp chí Gartner.
- ❖ 2004: Google phát hành Hadoop, một hệ thống phân tán mã nguồn mở cho xử lý dữ liệu lớn.

- ❖ 2005: Apache Spark được phát triển, là một nền tảng xử lý dữ liệu lớn thời gian thực và phi thời gian thực.
- ❖ 2010: Các công nghệ điện toán đám mây (cloud computing) ra đời, cung cấp nền tảng cho việc lưu trữ và xử lý dữ liệu lớn trên quy mô lớn.
- Giai đoạn hiện đại (2010-nay)

Giai đoạn này đánh dấu sự phát triển mạnh mẽ của Big Data, với sự ra đời của các công nghệ và phương pháp xử lý dữ liệu lớn mới, bao gồm:

- ❖ 2012: Chính phủ Hoa Kỳ tuyên bố Sáng kiến Nghiên cứu và Phát triển Dữ liệu lớn (Big Data Research and Development Initiative).
- ❖ 2013: Các công ty công nghệ lớn như Amazon, Microsoft và IBM bắt đầu cung cấp các dịch vụ xử lý dữ liệu lớn trên nền tảng điện toán đám mây.
- ❖ 2014: Các công nghệ học máy và trí tuệ nhân tạo (AI) được ứng dụng rộng rãi trong xử lý dữ liệu lớn.
- ❖ 2020: Đại dịch COVID-19 đã thúc đẩy sự phát triển của Big Data, khi các doanh nghiệp và tổ chức cần sử dụng Big Data để phân tích dữ liệu từ các nguồn khác nhau để đưa ra các quyết định kinh doanh và ứng phó với đại dịch.

### **1.3. Công nghệ và công cụ xử lý dữ liệu lớn:**

Công nghệ và công cụ xử lý dữ liệu lớn là những công nghệ và phần mềm được sử dụng để thu thập, lưu trữ, xử lý và phân tích dữ liệu lớn. Dữ liệu lớn là tập dữ liệu có khối lượng lớn, tốc độ nhanh và tính đa dạng cao. Các công nghệ và công cụ xử lý dữ liệu lớn giúp các tổ chức và cá nhân có thể khai thác giá trị từ dữ liệu lớn, từ đó đưa ra những quyết định sáng suốt hơn.

Có thể phân loại các công nghệ và công cụ xử lý dữ liệu lớn thành ba nhóm chính:

- Công nghệ thu thập dữ liệu: Các công nghệ này được sử dụng để thu thập dữ liệu từ các nguồn khác nhau, bao gồm:
  - + Các thiết bị cảm biến
  - + Các hệ thống máy tính
  - + Các ứng dụng web
  - + Các phương tiện truyền thông xã hội

- Công nghệ lưu trữ dữ liệu: Các công nghệ này được sử dụng để lưu trữ dữ liệu lớn một cách hiệu quả và an toàn, bao gồm:

- + Các cơ sở dữ liệu phân tán
- + Các hệ thống lưu trữ đám mây
- + Các công nghệ lưu trữ băng từ

– Công nghệ xử lý dữ liệu: Các công nghệ này được sử dụng để xử lý dữ liệu lớn, bao gồm:

- + Các thuật toán học máy
- + Các công nghệ khai thác dữ liệu
- + Các công nghệ phân tích dữ liệu

- Có nhiều công nghệ và công cụ được phát triển để xử lý dữ liệu lớn (Big Data) hiệu quả. Dưới đây là một số công nghệ và công cụ phổ biến trong lĩnh vực này:

+ Hadoop: Apache Hadoop là một framework mã nguồn mở được sử dụng để xử lý và lưu trữ dữ liệu lớn phân tán trên các cụm máy tính. Nó cung cấp cơ chế xử lý phân tán (distributed processing) và hệ thống tệp phân tán (distributed file system) giúp xử lý dữ liệu lớn một cách hiệu quả.

+ Spark: Apache Spark là một framework mã nguồn mở cung cấp khả năng xử lý dữ liệu lớn với tốc độ nhanh hơn so với Hadoop. Spark hỗ trợ xử lý phân tán, tính toán trên bộ nhớ và cung cấp các API cho việc xử lý dữ liệu lớn, bao gồm xử lý batch và xử lý thời gian thực.

+ NoSQL databases: Các cơ sở dữ liệu NoSQL (Not Only SQL) được thiết kế để làm việc với dữ liệu phi cấu trúc và có khả năng mở rộng tốt. Các hệ thống NoSQL như MongoDB, Cassandra, và Redis được sử dụng để lưu trữ và truy vấn dữ liệu lớn một cách linh hoạt và hiệu quả.

+ Công cụ trực quan hóa dữ liệu: Công cụ trực quan hóa dữ liệu như Tableau, Power BI, và D3.js giúp biểu đồ hóa và trực quan hóa dữ liệu lớn để dễ dàng hiểu và phân tích. Chúng cung cấp các biểu đồ, đồ thị, và bảng tin tức để trực quan hóa dữ liệu một cách hấp dẫn và trực quan.

+ Công nghệ xử lý dữ liệu thời gian thực: Công nghệ xử lý dữ liệu thời gian thực như Apache Kafka, Apache Flink, và Apache Storm được sử dụng để xử lý dữ liệu lớn trong thời gian thực. Chúng hỗ trợ việc xử lý luồng dữ liệu (stream processing) và phân tích dữ liệu thời gian thực để phản ứng nhanh chóng với dữ liệu đang chảy.

+ Công cụ phân tích dữ liệu và học máy: Các công cụ như Python, Scala với các thư viện như NumPy, Pandas, và scikit-learn cung cấp các công cụ phân tích dữ liệu và học máy mạnh mẽ để khám phá, phân tích và rút trích tri thức từ dữ liệu lớn.

- Ở đây chúng em sử dụng Apache Flink để thực hiện đề tài này “Tìm Hiểu Về Apache Flink Và Ứng Dụng Trong Bài Toán Xử Lý Dữ Liệu Theo Luồng”.

#### **1.4. Tầm quan trọng và ảnh hưởng của dữ liệu lớn:**

- Dữ liệu lớn mang lại nhiều lợi ích và tiềm năng cho các tổ chức và doanh nghiệp. Việc khai thác và phân tích dữ liệu lớn giúp tạo ra thông tin hữu ích và những hiểu biết sâu sắc về người dùng, khách hàng và hoạt động kinh doanh. Điều này giúp nâng cao quyết định, tăng cường hiệu suất và đưa ra chiến lược cạnh tranh. Ngoài ra, dữ liệu lớn cũng đóng góp vào việc nghiên cứu khoa học, phát triển sản phẩm và dịch vụ mới.

#### **1.5. Lợi ích và thách thức của xử lý dữ liệu lớn:**

- Lợi ích: Xử lý dữ liệu lớn mang lại nhiều lợi ích và ứng dụng đa dạng trong nhiều lĩnh vực khác nhau. Các lĩnh vực chủ chốt bao gồm:

+ Kinh doanh và tiếp thị: Xử lý dữ liệu lớn giúp hiểu khách hàng, dự đoán xu hướng và hành vi tiêu dùng, tối ưu hóa chiến lược tiếp thị và tăng cường trải nghiệm khách hàng.

+ Y tế: Dữ liệu lớn được sử dụng để phân tích thông tin y tế, nghiên cứu bệnh tật, dự báo dịch bệnh và tăng cường chẩn đoán và điều trị bệnh.

+ Tài chính: Xử lý dữ liệu lớn giúp phân tích tài chính, dự báo rủi ro, giao dịch tài chính nhanh chóng và xây dựng hệ thống phân tích rủi ro.

+ Giao thông và vận tải: Dữ liệu lớn được sử dụng để tối ưu hóa lưu lượng giao thông, dự đoán nhu cầu và cải thiện hệ thống vận chuyển.

- Thách thức:

+ Khối lượng dữ liệu: Dữ liệu lớn có khối lượng rất lớn, đòi hỏi hệ thống phải có khả năng lưu trữ và xử lý dữ liệu với tốc độ cao.

+ Đa dạng dữ liệu: Dữ liệu lớn có tính đa dạng, bao gồm dữ liệu cấu trúc, bất cấu trúc và bán cấu trúc. Điều này đòi hỏi công cụ và phương pháp phù hợp để khai thác và phân tích các loại dữ liệu này.

+ Tính tương tự và phân tán: Dữ liệu lớn thường được phân tán trên nhiều nguồn, đòi hỏi các giải pháp phân tán và công nghệ xử lý phức tạp để khám phá và kết nối các nguồn dữ liệu.

+ Bảo mật và quyền riêng tư: Dữ liệu lớn thường chứa thông tin nhạy cảm, đòi hỏi sự quản lý bảo mật và tuân thủ các quy định về quyền riêng tư.

- Việc giải quyết các thách thức này đòi hỏi sự phát triển của công nghệ và phương pháp xử lý dữ liệu lớn, cũng như việc xây dựng cơ sở hạ tầng phù hợp để lưu trữ và xử lý dữ liệu lớn một cách hiệu quả.

## CHƯƠNG 2: GIỚI THIỆU ĐỀ TÀI

### 2.1. Giới thiệu chung về Apache Flink

#### 2.1.1. Khái niệm Apache Flink



Hình 2.1 Hình ảnh Apache Flink

- Apache Flink là một hệ thống xử lý dữ liệu phân tán mã nguồn mở. Nó được phát triển bởi Apache Software Foundation và được thiết kế để xử lý dữ liệu với tốc độ cao, độ chính xác và khả năng mở rộng. Cốt lõi của Apache Flink là một hệ thống xử lý dòng dữ liệu phân tán được viết bằng Java và Scala.

- Flink thực thi các chương trình dòng dữ liệu một cách song song và tuần tự, cho phép xử lý hàng loạt và xử lý dòng cùng một lúc. Hơn nữa, hệ thống chạy tuần tự của Flink hỗ trợ thực thi các thuật toán lặp một cách tự nhiên.

- Apache Flink là một framework dùng để xử lý luồng dữ liệu từ bounded streams và unbounded streams. Nó có thể chạy hầu hết trên mọi môi trường cluster phổ biến hiện nay (Kubernetes, YARN,...) và thực hiện tính toán luồng dữ liệu với tốc độ rất nhanh.

- Apache Flink, tương tự Spark, là một hệ thống xử lý phân tán và xử lý dữ liệu khối (batch processing). Thành phần cốt lõi của Flink cung cấp tính năng phân phối dữ liệu, kết nối và tăng khả năng chịu lỗi cho các máy chủ trong hệ thống cluster.

- Hệ thống xử lý chính của kiến trúc Flink cần được nói đến chính là DataStreams. Các nguồn dữ liệu mà Flink có thể tiếp nhận đến từ một hệ thống Kafka, Twitter, và ZeroMQ. Nguồn dữ liệu sau khi xử lý được ghi vào tập tin dữ liệu hoặc ghi thông qua giao thức socket. Việc hoạt động của Flink tiến hành qua cơ chế JVM trên máy cục bộ hoặc xử lý theo kiến trúc cluster. Quá trình chuyển đổi dữ liệu được Flink hỗ trợ trên DataStreams bao gồm: Map, FlatMap, Filter, Reduce, Fold, Aggregations, Window, WindowAll,

Window Reduce, Window Fold, Window Join, Window CoGroup, Split, và một số dạng khác.

- Ứng dụng DataStream xử lý liên tục và trong thời gian dài, Flink cung cấp cơ chế giảm thiểu khả năng lỗi hệ thống thông qua cơ chế Lightweight Distributed Snapshot; dựa theo phương thức hoạt động của Chandy-Lamport distributed snapshot. Flink vẫn có thể vẫn tiếp tục các quá trình tính toán dữ liệu trong khi tiến hành các bảo trì hệ thống. Flink thực hiện kiểm tra trạng thái dữ liệu để bảo đảm nó có thể phục hồi khi xuất hiện hư hỏng trong dữ liệu.

- DataStream API hỗ trợ chức năng chuyển đổi (transformation) trên các luồng dữ liệu thông qua cơ chế cửa sổ (window). Người dùng có thể tùy biến kích thước window, tần suất tiếp nhận dữ liệu hoặc các phương thức gọi dữ liệu. Window có thể hoạt động dựa trên nhiều chính sách điều khiển như count, time, và delta.

- Apache Flink có những đối tượng tương tự Apache Spark, nhưng khác nhau cơ bản về kiến trúc thiết kế. Flink thể hiện tính ưu việt trên chính kiến trúc và khả năng xử lý của nó; bao gồm batch, micro-batch, và xử lý sự kiện riêng lẻ, tất cả chỉ trong một hệ thống duy nhất.

### **2.1.2. Lịch sử hình thành và phát triển**

- Flink ban đầu được gọi Stratosphere và phát triển tại Đại học Kỹ thuật Berlin. Nó bây giờ đã trở thành một thay thế cho cả MapReduce và Spark. Một số lợi ích của Apache Flink bao gồm khả năng sử dụng các thuật toán tương tự cho luồng và hàng loạt chế độ, để thúc đẩy những thành tựu độ trễ thấp.

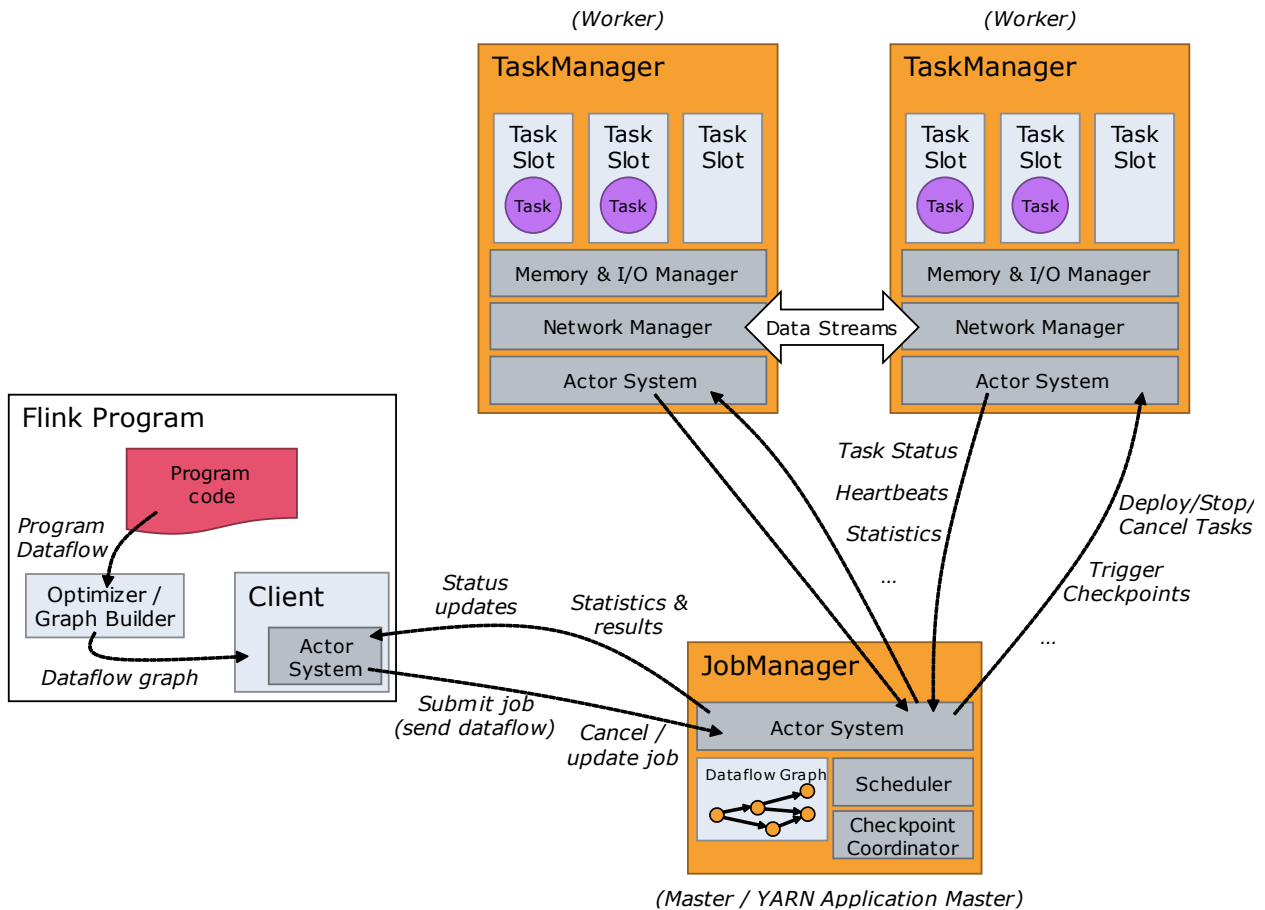
- Ngoài ra còn có cơ hội để phân tích cụm và thực hiện xử lý lặp đi lặp lại trên các nút khác nhau của hệ thống cùng một lúc. Apache Flink cũng tương thích với Apache S3 và các công cụ khác Hadoop liên quan. công cụ châu Âu phát triển này có thể sẽ trở thành một phần trong những cách cụ thể mà nhà quản trị xử lý hệ thống dữ liệu lớn như các công ty nhảy trên bandwagon cho phân tích dữ liệu có dung lượng lớn.

## **2.2. Kiến trúc của Apache Flink**

### **2.2.1. Kiến trúc**



Kiến trúc của Flink bao gồm một hệ thống phân tán được sử dụng để xử lý luồng dữ liệu, gồm hai loại tiến trình chính là JobManager và TaskManagers và các thành phần phụ khác như ResourceManager, Dispatcher và JobMaster.



Hình 2.2 Tiến trình được sử dụng trong việc thực thi một chương trình Flink

Hình trên minh họa các tiến trình được thực thi trong flink nó cho thấy kiến trúc của Flink bao gồm một JobManager và một hoặc nhiều TaskManager. JobManager quản lý các tài nguyên và lên lịch thực thi các tác vụ, trong khi TaskManager thực hiện các tác vụ và trao đổi dữ liệu. Hình ảnh cũng cho thấy rằng một ứng dụng Flink có thể được chạy trên nhiều tiến trình TaskManager khác nhau, tùy thuộc vào số lượng tác vụ cần được thực hiện và mức độ song song của chúng.

### 2.2.1.1. JobManager

JobManager trong Apache Flink là thành phần quan trọng trong kiến trúc của hệ thống. Nhiệm vụ chính của JobManager là quản lý và điều phối các công việc (job) trong cụm Flink. Khi nhận được một công việc, JobManager phân chia nó thành các tác vụ nhỏ

hơn và giao cho các TaskManager thực thi. Nó quản lý thông tin về các công việc đang chạy, bao gồm trạng thái của các tác vụ, lập lịch và phân phối tài nguyên cho các tác vụ, quản lý việc điều phối dữ liệu giữa các tác vụ và quản lý trạng thái của các công việc. JobManager đóng vai trò quan trọng trong việc đảm bảo hiệu suất và đáng tin cậy của hệ thống Flink.

JobManager khởi tạo các thành phần nội bộ, bao gồm:

- Resource manager: quản lý tài nguyên hệ thống (CPU, memory, network) cho các TaskManager.
- Dispatcher: nhận các JobGraph (biểu đồ mô tả ứng dụng Flink) từ client và phân tích chúng.
- Scheduler: lên lịch thực thi các task dựa trên JobGraph và tài nguyên khả dụng.
- Archive: lưu trữ các thông tin về các job đã thực thi.
- High Availability Services (HA): đảm bảo tính sẵn sàng cao của JobManager, cho phép phục hồi sau sự cố.

### **2.2.1.2. TaskManagers**

TaskManager (còn được gọi là workers) thực thi các tác vụ của một luồng dữ liệu, và đệm và trao đổi các luồng dữ liệu.

Luôn phải có ít nhất một TaskManager. Đơn vị nhỏ nhất của lập lịch tài nguyên trong một TaskManager là một khe tác vụ. Số lượng khe tác vụ trong một TaskManager cho biết số lượng tác vụ xử lý đồng thời. Lưu ý rằng nhiều toán tử có thể được thực thi trong một khe tác vụ.

#### ❖ Một số điểm quan trọng về TaskManagers:

Thực thi tác vụ: TaskManagers là nơi thực hiện các tác vụ xử lý dữ liệu trong Flink. Khi nhận được các tác vụ từ JobManager, TaskManagers thực hiện tính toán, xử lý và biến đổi dữ liệu theo yêu cầu của công việc. Các tác vụ có thể chạy song song trên nhiều TaskManagers, tận dụng sức mạnh tính toán của cụm.

Quản lý tài nguyên: TaskManagers quản lý và sử dụng tài nguyên tính toán trên nút của chúng. Điều này bao gồm CPU, bộ nhớ và dung lượng đĩa. TaskManagers cung cấp một môi trường thực thi cho các tác vụ và đảm bảo rằng tài nguyên cần thiết được phân phối và sử dụng một cách hiệu quả.

Giao tiếp với JobManager: TaskManagers giao tiếp với JobManager để nhận các tác vụ mới và báo cáo trạng thái thực thi của chúng. Chúng thông báo về tiến trình và kết quả của các tác vụ, cho phép JobManager theo dõi và quản lý công việc trong cụm.

Quản lý trạng thái: TaskManagers có khả năng quản lý và lưu trữ trạng thái của tác vụ trong quá trình thực thi. Điều này cho phép Flink xử lý các tác vụ có trạng thái và duy trì trạng thái của chúng qua các giai đoạn xử lý khác nhau. Quản lý trạng thái giúp Flink đạt được tính nhất quán và khả năng chịu lỗi.

Gửi và nhận dữ liệu: TaskManagers có khả năng nhận dữ liệu từ nguồn dữ liệu và gửi kết quả đến các đích dữ liệu. Chúng có thể truy cập các hệ thống lưu trữ dữ liệu như Apache Kafka, Apache Cassandra hoặc HDFS. Điều này cho phép Flink tích hợp dữ liệu từ nhiều nguồn khác nhau và đẩy kết quả đến các hệ thống lưu trữ hoặc hệ thống tiêu thụ khác.

### 2.2.2. Nguyên lý hoạt động

Apache Flink là một nền tảng xử lý dữ liệu phân tán, có thể được sử dụng để xử lý dữ liệu luồng và dữ liệu batch. Flink hoạt động theo nguyên tắc sau:

- Tập trung (centralized): Flink có một thành phần trung tâm gọi là JobManager chịu trách nhiệm khởi tạo, điều phối và giám sát việc thực thi các ứng dụng Flink.
  - Phân tán (distributed): Các ứng dụng Flink được chia thành các task, mỗi task được thực thi trên một máy chủ TaskManager.
  - Tính trạng thái (stateful): Flink có thể xử lý dữ liệu có trạng thái, tức là dữ liệu cần được lưu trữ giữa các lần xử lý.
- Quy trình hoạt động của Apache Flink có thể được chia thành các bước sau:
- Khởi tạo (initialization): JobManager được khởi tạo và bắt đầu lắng nghe các yêu cầu từ các ứng dụng Flink.
  - Gửi JobGraph (submitting JobGraph): Ứng dụng Flink gửi JobGraph đến JobManager. JobGraph là một biểu đồ mô tả cấu trúc của ứng dụng Flink.
  - Phân tích JobGraph (analyzing JobGraph): JobManager phân tích JobGraph để xác định các task cần thực thi và các tài nguyên cần thiết.
  - Phân phối task (distributing tasks): JobManager phân phối các task đến các TaskManager có sẵn.

- Thực thi task (executing tasks): Các TaskManager thực thi các task theo thứ tự được xác định trong JobGraph.
- Giám sát (monitoring): JobManager giám sát tiến trình thực thi của các task và phát hiện lỗi.
- Kết thúc (terminating): Sau khi tất cả các task hoàn thành, JobManager kết thúc ứng dụng Flink.

### 2.3 Tầm quan trọng của Apache Flink

Apache Flink có tầm quan trọng lớn trong lĩnh vực xử lý dữ liệu phân tán và phân tích thời gian thực.

- Xử lý dữ liệu phân tán: Flink cho phép xử lý dữ liệu phân tán trên một cụm máy tính, giúp tận dụng được khả năng tính toán và lưu trữ phân tán. Điều này cần thiết khi làm việc với dữ liệu lớn và yêu cầu hiệu suất cao.

- Xử lý dữ liệu stream thời gian thực: Flink là một trong những công nghệ hàng đầu cho việc xử lý dữ liệu stream thời gian thực. Nó hỗ trợ xử lý dữ liệu theo thời gian sự kiện (event time) và khắc phục được các vấn đề như trễ, độ trễ không đều, và sự cố mạng. Điều này rất quan trọng trong các ứng dụng như phân tích thời gian thực, giao dịch tài chính, đám mây dữ liệu, IoT và nhiều lĩnh vực khác.

- Tính toán trạng thái: Apache Flink hỗ trợ tính toán trạng thái phức tạp và theo dõi thông tin liên tục trong quá trình xử lý dữ liệu. Điều này cho phép bạn thực hiện các phân tích phức tạp, tính toán liên tục và ghi nhớ trạng thái của các sự kiện trước đó.

- Tính bền vững trong trường hợp lỗi: Flink cung cấp tính năng đáng tin cậy và bền vững trong trường hợp lỗi. Với cơ chế checkpointing, nó có thể sao lưu trạng thái công việc và dữ liệu, giúp đảm bảo tính nhất quán và khả năng khôi phục sau khi xảy ra sự cố.

- Tích hợp với hệ sinh thái Apache: Flink tích hợp tốt với các công cụ và hệ thống khác trong hệ sinh thái Apache như Apache Kafka, Apache Hadoop, Apache Hive, và nhiều công nghệ khác. Điều này tạo ra sự linh hoạt và khả năng mở rộng cho việc tích hợp và triển khai các giải pháp phân tích dữ liệu phức tạp.

### 2.4 Tính năng

Apache Flink cung cấp một loạt tính năng quan trọng để xử lý dữ liệu phân tán và phân tích thời gian thực.

- Xử lý dữ liệu stream và hàng loạt: Flink hỗ trợ xử lý cả dữ liệu stream và hàng loạt. Điều này cho phép người dùng xử lý dữ liệu đến từ các nguồn dữ liệu liên tục cũng như dữ liệu được lưu trữ trong các kho dữ liệu hàng loạt.

- Xử lý dữ liệu theo thời gian sự kiện: Flink hỗ trợ xử lý dữ liệu theo thời gian sự kiện (event time), cho phép xử lý dữ liệu dựa trên thời gian xảy ra của sự kiện thực tế. Điều này đặc biệt hữu ích khi làm việc với dữ liệu ghi nhận từ các hệ thống theo thời gian thực như IoT và giao dịch tài chính.

- Tính toán trạng thái và ghi nhớ: Flink cho phép tính toán trạng thái phức tạp và theo dõi thông tin liên tục trong quá trình xử lý dữ liệu. Điều này giúp thực hiện các phân tích phức tạp, tính toán liên tục và theo dõi trạng thái của các sự kiện trước đó.

- Fault Tolerance (Sự bền vững trong trường hợp lỗi): Flink cung cấp tính năng bền vững trong trường hợp lỗi thông qua cơ chế checkpointing. Các checkpoint được sử dụng để sao lưu trạng thái của công việc và dữ liệu, đảm bảo khả năng khôi phục sau khi xảy ra sự cố.

- Thời gian thực và độ trễ thấp: Flink được thiết kế để xử lý dữ liệu thời gian thực với độ trễ thấp. Nó cung cấp khả năng xử lý sự kiện ngay khi chúng đến và đáp ứng nhanh chóng với dữ liệu mới nhất.

- Tích hợp với hệ sinh thái Apache: Flink tích hợp tốt với các công cụ và hệ thống khác trong hệ sinh thái Apache như Apache Kafka, Apache Hadoop, Apache Hive và nhiều công nghệ khác. Điều này tạo ra sự linh hoạt và khả năng mở rộng cho việc tích hợp và triển khai các giải pháp phân tích dữ liệu phức tạp.

- Hỗ trợ ngôn ngữ lập trình đa dạng: Flink hỗ trợ nhiều ngôn ngữ lập trình như Java, Scala và Python. Điều này cho phép người dùng sử dụng ngôn ngữ mà họ thoải mái nhất để triển khai ứng dụng Flink.

## 2.5. Lợi ích

Xử lý dữ liệu trong thời gian thực: Apache Flink cho phép các ứng dụng xử lý dữ liệu trực tiếp khi nó được tạo ra hoặc nhận được, giúp cho các ứng dụng này có thể đưa ra các quyết định ngay lập tức.

Xử lý dữ liệu phân tán: Apache Flink cung cấp tính năng xử lý dữ liệu phân tán, cho phép các ứng dụng xử lý dữ liệu lớn và phức tạp.

**Tính sẵn sàng cao:** Apache Flink cung cấp tính năng đảm bảo tính sẵn sàng hoạt động của hệ thống trong trường hợp có lỗi xảy ra, giúp cho các ứng dụng có thể hoạt động liên tục và đáp ứng được nhu cầu của người dùng.

**Hiệu suất cao:** Apache Flink được thiết kế để đạt hiệu suất cao trong việc xử lý dữ liệu, giúp cho các ứng dụng có thể hoạt động nhanh chóng và hiệu quả.

**Dễ sử dụng:** Apache Flink cung cấp các API và công cụ dễ sử dụng để phát triển ứng dụng xử lý dữ liệu, giúp cho các nhà phát triển có thể phát triển các ứng dụng một cách dễ dàng và nhanh chóng.

**Cộng đồng lớn:** Apache Flink được phát triển bởi một cộng đồng lớn các nhà phát triển và người dùng trên toàn thế giới, giúp cho nó luôn được cập nhật và phát triển theo thời gian, để đáp ứng được các yêu cầu của người sử dụng.

**Hỗ trợ đa nền tảng:** Apache Flink hỗ trợ các nền tảng xử lý dữ liệu phân tán phổ biến như Hadoop và Kubernetes.

**Tích hợp với các công cụ khác:** Apache Flink có thể tích hợp với các công cụ khác như Apache Kafka, Apache Cassandra, Elasticsearch và nhiều hệ thống lưu trữ dữ liệu khác.

## CHƯƠNG 3: CÀI ĐẶT VÀ THAO TÁC APACHE FLINK

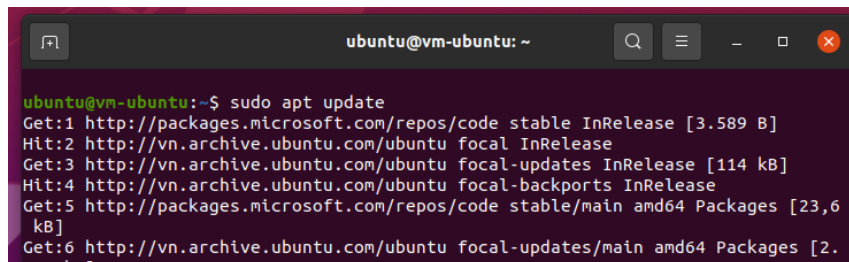
### 3.1. Cài đặt Apache Flink trên Ubuntu 20.04 LTS

- + Java Development Kit (JDK): Bạn cần cài đặt JDK trước khi cài đặt Apache Flink. Phiên bản khuyến nghị là JDK 11 hoặc cao hơn.
- + Hệ điều hành Ubuntu 20.04 LTS: Đảm bảo bạn đang chạy Ubuntu 20.04 LTS hoặc phiên bản tương đương.
- + Quyền quản trị: Để cài đặt và cấu hình Flink, bạn cần có quyền quản trị trên hệ thống Ubuntu.
- Cấu hình tối thiểu:
  - + RAM: Tối thiểu 4GB RAM. Điều này đảm bảo bạn có đủ dung lượng RAM để chạy Apache Flink cùng với các ứng dụng liên quan.
  - + CPU: Tối thiểu 2 CPU cores. Điều này giúp đảm bảo hiệu suất ổn định và xử lý thông điệp một cách nhanh chóng.

#### Bước 1: Cập nhật hệ thống

Các lệnh này sẽ tải xuống và cài đặt các bản cập nhật mới nhất cho hệ thống.

`sudo apt update`



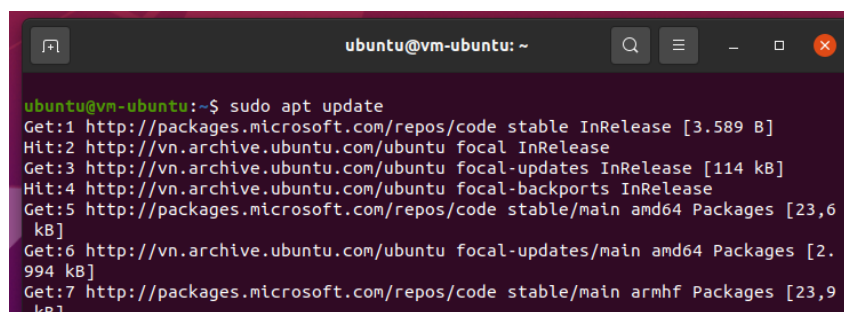
```

ubuntu@vm-ubuntu: ~
ubuntu@vm-ubuntu:~$ sudo apt update
Get:1 http://packages.microsoft.com/repos/code stable InRelease [3.589 B]
Hit:2 http://vn.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://vn.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:4 http://vn.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:5 http://packages.microsoft.com/repos/code stable/main amd64 Packages [23,6 kB]
Get:6 http://vn.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2.994 kB]

```

Hình 3.1 Hình ảnh chạy lệnh update

`sudo apt upgrade`



```

ubuntu@vm-ubuntu: ~
ubuntu@vm-ubuntu:~$ sudo apt update
Get:1 http://packages.microsoft.com/repos/code stable InRelease [3.589 B]
Hit:2 http://vn.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://vn.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:4 http://vn.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:5 http://packages.microsoft.com/repos/code stable/main amd64 Packages [23,6 kB]
Get:6 http://vn.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2.994 kB]
Get:7 http://packages.microsoft.com/repos/code stable/main armhf Packages [23,9 kB]

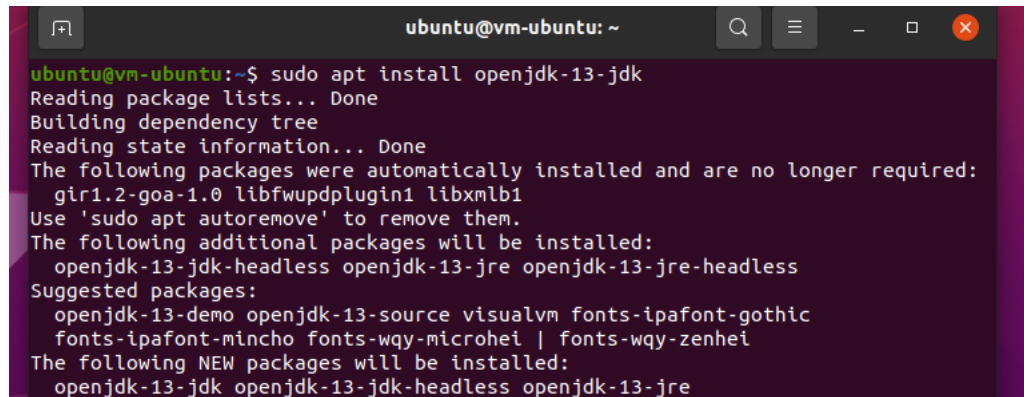
```

Hình 3.2 Hình ảnh chạy lệnh upgrade

## Bước 2: Cài đặt các gói cần thiết

Apache Flink yêu cầu Java từ phiên bản 11 để chạy.

```
sudo apt install openjdk-13-jdk
```



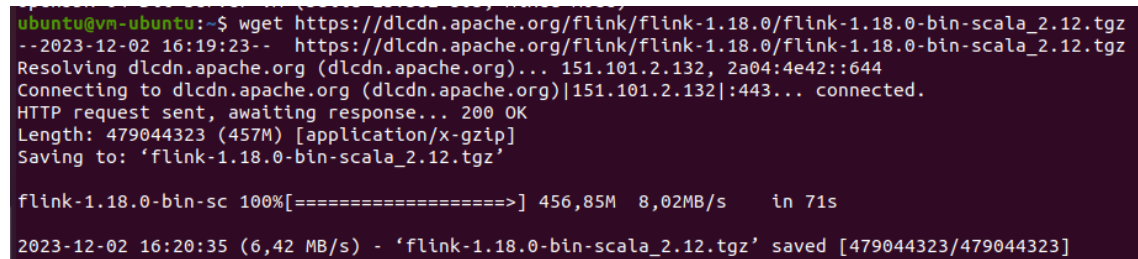
```
ubuntu@vm-ubuntu: ~$ sudo apt install openjdk-13-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 libfwupdplugin1 libxmlb1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  openjdk-13-jdk-headless openjdk-13-jre openjdk-13-jre-headless
Suggested packages:
  openjdk-13-demo openjdk-13-source visualvm fonts-ipafont-gothic
  fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei
The following NEW packages will be installed:
  openjdk-13-jdk openjdk-13-jdk-headless openjdk-13-jre
```

Hình 3.3 Hình ảnh chạy lệnh install jdk

## Bước 3: Tải xuống Apache Flink

Tải flink về (version mới nhất là 1.18.0)

```
wget https://dlcdn.apache.org/flink/flink-1.18.0/flink-1.18.0-bin-scala_2.12.tgz
```



```
ubuntu@vm-ubuntu:~$ wget https://dlcdn.apache.org/flink/flink-1.18.0/flink-1.18.0-bin-scala_2.12.tgz
--2023-12-02 16:19:23-- https://dlcdn.apache.org/flink/flink-1.18.0/flink-1.18.0-bin-scala_2.12.tgz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 479044323 (457M) [application/x-gzip]
Saving to: 'flink-1.18.0-bin-scala_2.12.tgz'

flink-1.18.0-bin-sc 100%[=====] 456,85M 8,02MB/s in 71s

2023-12-02 16:20:35 (6,42 MB/s) - 'flink-1.18.0-bin-scala_2.12.tgz' saved [479044323/479044323]
```

Hình 3.4 Hình ảnh chạy lệnh tải Apache Flink

Sau đó giải nén file:

```
tar xzf flink-*.tgz
```

Trở tới thư mục flink, mở start-cluster.sh

```
cd flink-1.18.0
```

```
./bin/start-cluster.sh
```



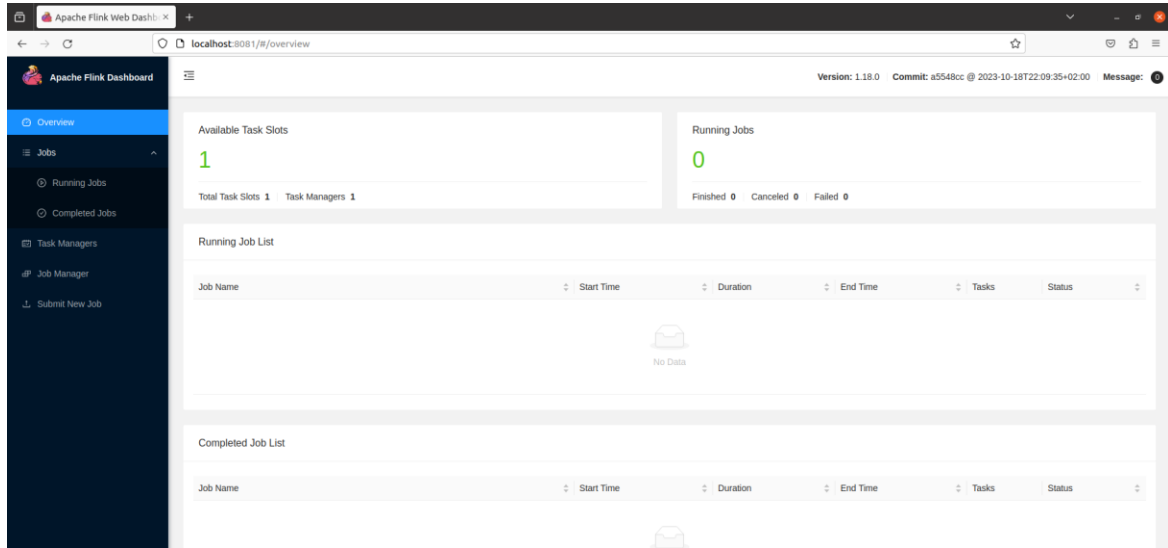
```
ubuntu@vm-ubuntu:~$ cd flink-1.18.0
ubuntu@vm-ubuntu:~/flink-1.18.0$ ./bin/start-cluster.sh
Starting cluster.
Starting standalone session daemon on host vm-ubuntu.
Starting taskexecutor daemon on host vm-ubuntu.
ubuntu@vm-ubuntu:~/flink-1.18.0$
```

Hình 3.5 Hình ảnh chạy lệnh khởi động Apache Flink



## Bước 4. Truy cập Giao diện web Apache Flink

- Sau khi cài đặt thành công, bây giờ hãy truy cập vào giao diện web của Apache Flink thông qua: **http://localhost:8081**

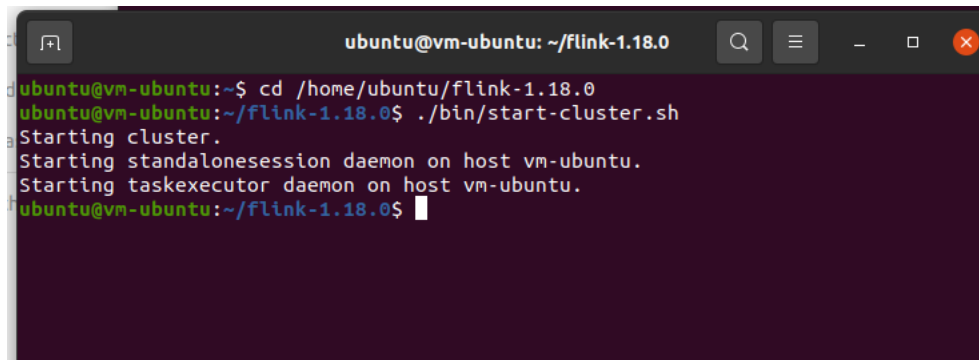


Hình 3.6 Hình ảnh giao diện Flink

## 3.2. Thao tác Apache Flink trên Ubuntu 20.04 LTS

### Bước 1. Khởi động Flink cluster

```
cd /home/ubuntu/flink-1.18.0 ./bin/start-cluster.sh
```



Hình 3.7 Hình ảnh chạy lệnh khởi động Apache Flink

**Bước 2. Mở một terminal khác và chạy lệnh sau để khởi động một socket server, lắng nghe kết nối đến cổng 9000 trên máy tính**

```
nc -lk -9000
```

```

ubuntu@vm-ubuntu: ~/flink-1.18.0
ubuntu@vm-ubuntu:~$ cd /home/ubuntu/flink-1.18.0
ubuntu@vm-ubuntu:~/flink-1.18.0$ nc -lk 9000

```

Hình 3.7 Hình ảnh chạy lệnh khởi động Socket

### Bước 3. Mở một terminal khác

./bin/flink run examples/streaming/SocketWindowWordCount.jar --hostname localhost --port 9000

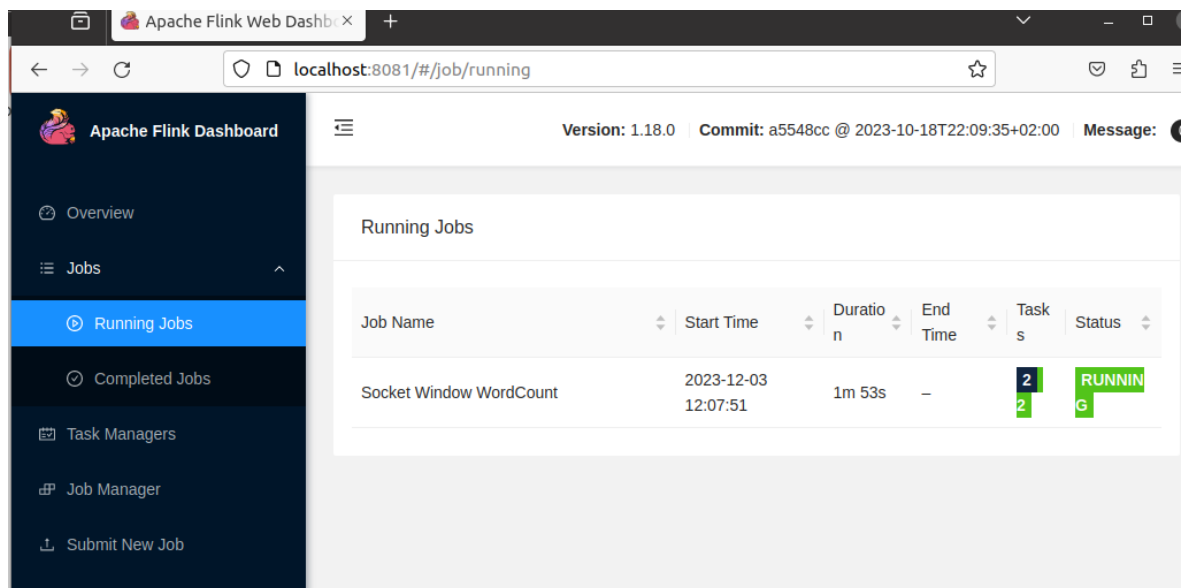
```

ubuntu@vm-ubuntu:~/flink-1.18.0$ ./bin/flink run examples/streaming/SocketWindowWordCount.jar --hostname localhost --port 9000
Job has been submitted with JobID 5e0a0ff6adc41d8c52083c5893211a4b

```

Hình 3.8 Hình ảnh chạy lệnh khởi động Wordcount

Qua giao diện localhost xem xem run lên chưa



Hình 3.9 Hình ảnh giao diện

Sau đó mở thêm 1 terminal mới, nhập dòng sau rồi enter

`tail -f log/flink-*-taskexecutor-*.out`

Tiếp đó, để song song 2 terminal để nhập và xem kết quả

```

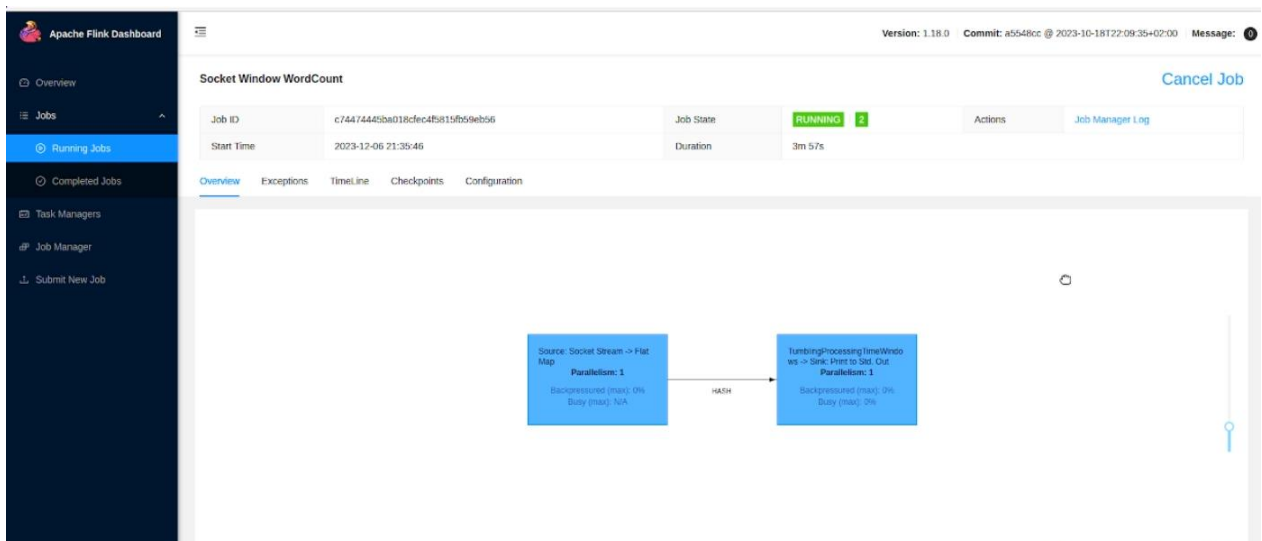
ubuntu@vm-ubuntu: ~/flink-1.18.0
ubuntu@vm-ubuntu:~$ cd /home/ubuntu/flink-1.18.0
ubuntu@vm-ubuntu:~/flink-1.18.0$ nc -lk 9000
yes
yes
no
no
ok
yes
yes
no
no
no
[]

ubuntu@vm-ubuntu: ~/flink-1.18.0
==> log/flink-ubuntu-taskexecutor-1-vm-ubuntu.out <==
==> log/flink-ubuntu-taskexecutor-0-vm-ubuntu.out <==
yes : 2
no : 2
ok : 1
yes : 1
yes : 1
no : 1
no : 1
no : 1
no : 1
no : 1

```

Hình 3.10 Hình ảnh demo

Hình ảnh web:



Hình 3.11 Hình ảnh web

## CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 4.1 Kết luận

#### Những điều làm được

- Đã cài đặt thực nghiệm bằng bài toán đếm từ
- Nắm được các kiến thức cơ bản về Apache Flink

#### Những điều chưa làm được

Vì thời gian và kiến thức có hạn, nhóm em đã cơ bản hoàn thành đề tài nhưng còn một vài hạn chế là chưa xây dựng hàm để xử lý bài toán theo luồng trên Apache Flink.

### 4.2 Hướng phát triển

- + Xử lý dữ liệu luồng dựa trên nhiều hàm hơn
- + Tự xây dựng hàm để xử lý bài toán theo luồng trên Apache Flink.

## TÀI LIỆU THAM KHẢO

- [1] Big Data là gì? Tất tần tât về Big Data: <https://topdev.vn/blog/big-data/>
- [2] Tổng quan về dữ liệu lớn:  
[https://vienthongke.vn/wpcontent/uploads/2021/04/Bai4.So5\\_.2016.pdf](https://vienthongke.vn/wpcontent/uploads/2021/04/Bai4.So5_.2016.pdf)
- [3] Apache Flink: [https://en.wikipedia.org/wiki/Apache\\_Flink](https://en.wikipedia.org/wiki/Apache_Flink)
- [4] Flink Architecture :<https://nightlies.apache.org/flink/flink-docs-master/docs/concepts/flink-architecture/>
- [5] Apache Flink - Quick guide : <https://viblo.asia/p/apache-flink-quick-guide-phan-1-924lJWoX5PM>