**1. Giới thiệu**

**1.1. Mục tiêu của báo cáo**

**Ingestion** là quá trình thu thập và chuyển dữ liệu từ nhiều nguồn (databases, APIs, file systems...) vào hệ thống xử lý. Trong kiến trúc dữ liệu hiện đại, ingestion là bước đầu tiên trong các pipeline dữ liệu.

**Orchestration** là quá trình điều phối, giám sát và tự động hóa các bước trong pipeline dữ liệu – thường là các bước ETL/ELT (Extract - Transform - Load).

**Tiêu chí lựa chọn công cụ:**

* Hỗ trợ ingestion dữ liệu từ nhiều nguồn.
* Tính năng orchestration mạnh mẽ: tự động hóa, retry, scheduling, logging.
* Khả năng mở rộng và tích hợp với hệ sinh thái hiện tại.
* Cộng đồng phát triển mạnh, tài liệu hỗ trợ đầy đủ.

**Lý do chọn công cụ:**

* **Apache NiFi**: mạnh về ingestion real-time và xử lý dữ liệu theo mô hình flow-based.
* **Dagster**: công cụ orchestration hiện đại, định nghĩa pipeline dưới dạng assets, tích hợp tốt với DevOps và CI/CD.

**1.2. Tổng quan về quá trình Ingestion và Orchestration**

* **Ingestion**: là khâu đầu của chuỗi giá trị dữ liệu, giúp đưa dữ liệu từ hệ thống nguồn vào hệ thống đích để xử lý và phân tích.
* **Orchestration**: kết nối các bước xử lý với nhau theo trình tự nhất định (ví dụ: lấy dữ liệu → làm sạch → lưu trữ). Công cụ orchestration giúp giảm thiểu lỗi và tăng tính tự động cho pipeline.

**2. Tổng quan về Apache NiFi**

**2.1. Giới thiệu và lịch sử phát triển**

* Apache NiFi bắt nguồn từ dự án "NiagaraFiles" do NSA phát triển, sau đó được open-source và gia nhập Apache Foundation năm 2014.
* Được thiết kế để xử lý dòng dữ liệu (data flows) một cách trực quan, dễ cấu hình.

**2.2. Kiến trúc và tính năng chính**

* **Flow-based Programming**: Giao diện thiết kế kéo-thả, thể hiện các bước xử lý dưới dạng khối và kết nối.
* **Ingestion mạnh mẽ**: Kết nối được với file system, API, Kafka, MQTT, FTP, v.v.
* **Orchestration & Automation**: Các processor có thể được cấu hình để chạy tự động, với khả năng branching, routing, retry logic, scheduling.
* **Clustering**: Hỗ trợ mở rộng theo cụm để xử lý lượng dữ liệu lớn.

**2.3. Ưu/nhược điểm**

* ✅ **Ưu điểm**:
  + Giao diện trực quan, dễ theo dõi.
  + Xử lý tốt dữ liệu streaming.
  + Dễ tích hợp nhiều nguồn dữ liệu.
* ❌ **Nhược điểm**:
  + Không lý tưởng cho pipeline phức tạp có nhiều logic điều kiện.
  + Tài liệu nâng cao còn thiếu.

**2.4. Tài liệu và nguồn tham khảo**

* Trang chủ: <https://nifi.apache.org>
* GitHub: <https://github.com/apache/nifi>

**3. Tổng quan về Dagster**

**3.1. Giới thiệu và lịch sử phát triển**

* Dagster là một open-source data orchestrator do Elementl phát triển.
* Hướng tới lập trình theo asset-based pipeline: tập trung vào dữ liệu được tạo ra chứ không chỉ là job.

**3.2. Kiến trúc và tính năng chính**

* **Asset-based pipeline**: xác định các "asset" (bảng dữ liệu, file...) và mối quan hệ giữa chúng.
* **Ingestion qua integration**: không thu thập dữ liệu trực tiếp như NiFi, nhưng có thể tích hợp với nguồn dữ liệu qua code (SQL, API, dbt...).
* **Orchestration mạnh mẽ**: Có scheduler, retry logic, conditional branching, logging, tích hợp CI/CD.
* **Tích hợp mạnh mẽ**: Docker, Kubernetes, dbt, Airbyte, Snowflake…

**3.3. Ưu/nhược điểm**

* ✅ **Ưu điểm**:
  + Dễ kiểm thử, dễ tích hợp DevOps.
  + Hệ sinh thái lập trình hiện đại.
  + Thích hợp cho team Data Engineer chuyên sâu.
* ❌ **Nhược điểm**:
  + Yêu cầu viết code – không có giao diện trực quan.
  + Cần thời gian làm quen với khái niệm assets và IO manager.

**3.4. Tài liệu và nguồn tham khảo**

* Trang chủ: <https://dagster.io>
* Tài liệu: https://docs.dagster.io

**4. So sánh chi tiết giữa Apache NiFi và Dagster**

**4.1. Tiêu chí so sánh**

| **Tiêu chí** | | **Apache NiFi** | **Dagster** |
| --- | --- | --- | --- |
| **tổng quan về tính năng và đặc điểm** | **Ingestion** | Kéo dữ liệu từ nhiều nguồn với giao diện kéo-thả | Tích hợp ingestion qua asset/script |
| **Orchestration** | Có scheduling và routing, nhưng giới hạn | Mạnh mẽ, hỗ trợ retry, conditional logic |
| **Khả năng tích hợp** | Clustering, REST API, Kafka, Hadoop | Tốt trong môi trường DevOps: Docker, dbt, Spark... |
| **Khả năng mở rộng** | Hỗ trợ clustering, scale-out tốt | Tốt, có thể triển khai trên Kubernetes, nhưng phức tạp hơn |
| **Độ phức tạp** | Dễ dùng cho người mới | Phù hợp với lập trình viên, phức tạp hơn |
| **CI/CD Support** | Có thể tích hợp nhưng không mạnh, thiếu tooling chuyên dụng | Có hỗ trợ |
| **Về hiệu suất** | **Xử lý real-time** | Mạnh, xử lý dòng dữ liệu theo thời gian thực | Chủ yếu xử lý batch, không tối ưu cho real-time |
| **Latency (độ trễ)** | Thấp với dữ liệu stream, tối ưu cho thời gian thực | Phụ thuộc scheduler, latency cao hơn trong batch |
| **Conditional branching** | Có, nhưng giới hạn và chủ yếu qua cấu hình UI | Có hỗ trợ |
| **Plugin/extension hỗ trợ** | Processor phong phú | qua Python lib |
| **Độ ổn định khi chạy lâu dài** | Ổn định cao trong môi trường production stream | Ổn định nếu cấu hình đúng, nhưng dễ gặp lỗi nếu CI/CD không tốt |
| **Quản lý lỗi (error handling)** | Hỗ trợ retry logic ở từng processor, log chi tiết | Retry logic mạnh, dễ kiểm thử nhưng yêu cầu viết code |

**4.2. Phân tích theo từng trường hợp sử dụng**

* **Apache NiFi**:

**+**  Thích hợp với những doanh nghiệp cần kết nối nhanh nhiều nguồn dữ liệu, yêu cầu real-time, không muốn viết nhiều code.

**+** NiFi dùng tốt với Hadoop ecosystem

* **Dagster**:

**+** Thích hợp với các hệ thống cần kiểm soát pipeline bằng code, yêu cầu kiểm thử và dễ dàng tích hợp CI/CD.

**+** Dagster phù hợp với modern stack (Snowflake, dbt, Airbyte...).

**5. Kết luận**

- Trong bối cảnh dữ liệu ngày càng đa dạng và khối lượng lớn, việc lựa chọn công cụ phù hợp để thực hiện Ingestion và Orchestration cho các pipeline dữ liệu là rất quan trọng.

- Apache NiFi và Dagster là hai công cụ đại diện cho hai cách tiếp cận khác nhau:

+ NiFi phù hợp với những hệ thống cần xử lý dữ liệu real-time, yêu cầu kết nối nhiều nguồn một cách linh hoạt mà không cần nhiều code. Với giao diện trực quan và khả năng kéo-thả, NiFi đặc biệt hữu ích cho các tổ chức cần triển khai nhanh các flow ingestion và xử lý streaming.

+ Dagster lại phù hợp với môi trường phát triển theo hướng lập trình hiện đại, đặc biệt khi pipeline cần được kiểm thử, kiểm soát chặt chẽ và tích hợp tốt vào quy trình CI/CD. Dagster hỗ trợ orchestration mạnh mẽ và cung cấp khả năng quản lý tài nguyên dữ liệu (assets) theo cách có cấu trúc và dễ bảo trì.

- Tùy theo nhu cầu và kiến trúc hệ thống, mỗi công cụ sẽ có điểm mạnh riêng:

+ Nếu mục tiêu là xử lý luồng dữ liệu thời gian thực và dễ dàng tích hợp với nhiều nguồn: Nên chọn Apache NiFi.

+ Nếu pipeline hướng tới batch processing, kiểm thử tự động, DevOps-friendly: Dagster là lựa chọn phù hợp hơn.

-Cả hai công cụ đều đang được phát triển mạnh mẽ, có cộng đồng hỗ trợ tốt và có thể mở rộng linh hoạt theo nhu cầu doanh nghiệp.