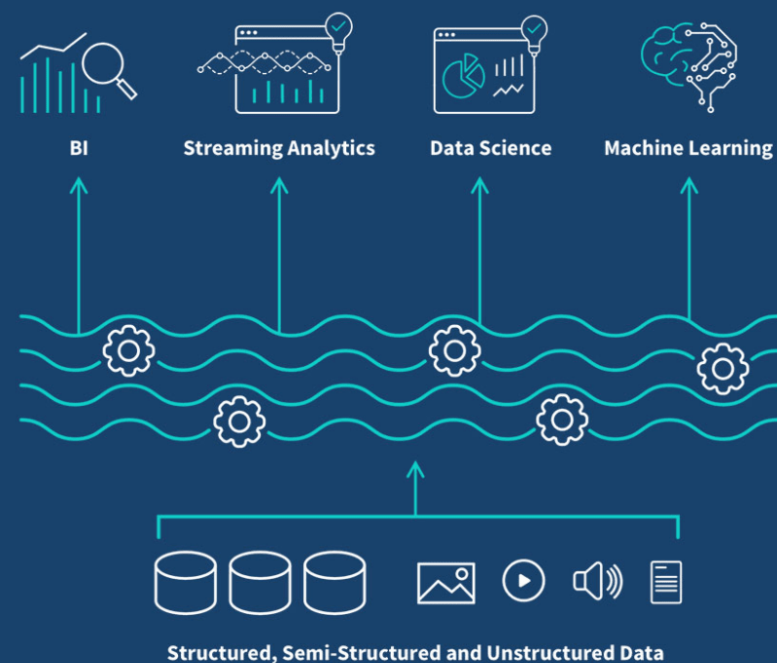


NGHIÊN CỨU VÀ TRIỂN KHAI HỆ THỐNG KIẾN TRÚC DỮ LIỆU DATA LAKEHOUSE CHO DOANH NGHIỆP

GVHD: TS. Nguyễn Mạnh Cường
Sinh viên: Bùi Quốc Triệu
Mã sinh viên: 2021600097



Cấu trúc

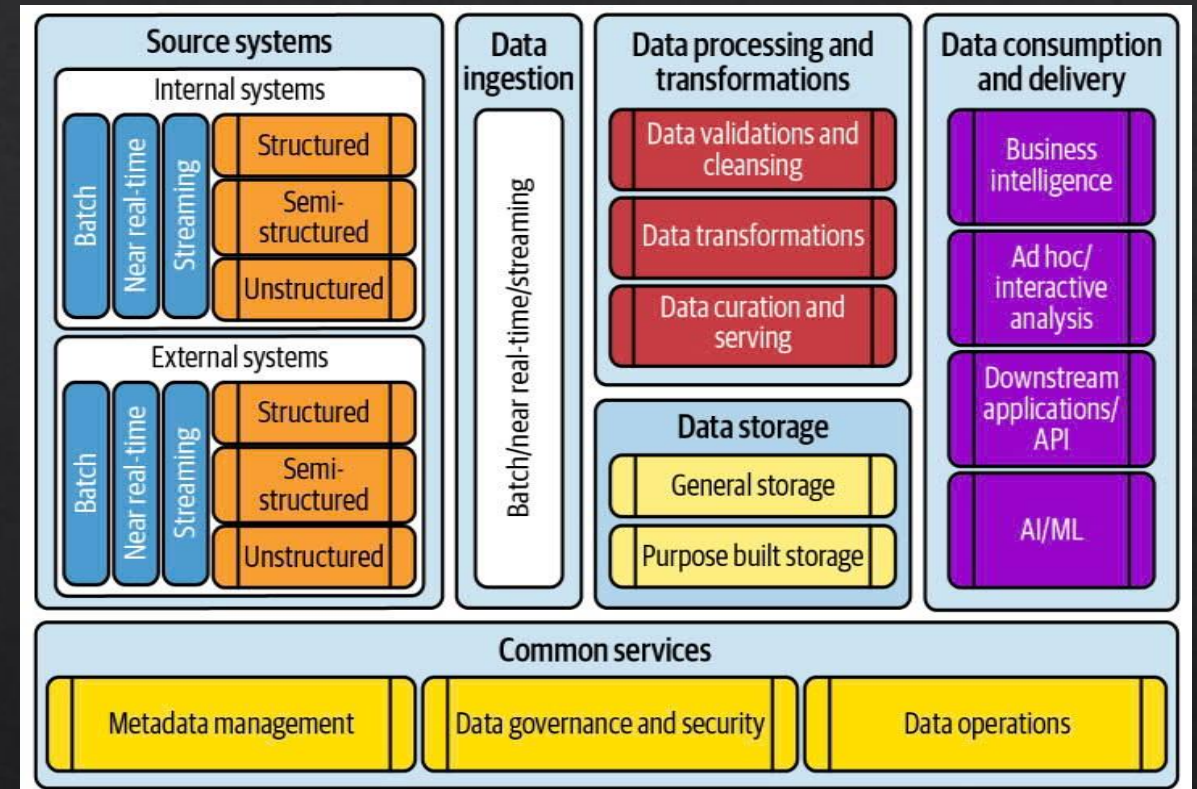
1. Tổng quan
2. Công nghệ sử dụng
3. Thực nghiệm

1. Tổng quan

- Hiện trạng: Công ty Viettel Network triển khai hệ thống dữ liệu lớn:
 - Môi trường tại chỗ (on-premises)
 - Quy mô 1000 máy chủ
 - Sử dụng công cụ triển khai tự động của Cloudera là CDH - Cloudera Hadoop. Công cụ này miễn phí và dễ dàng triển khai.
 - Nhiều nhược điểm:
 - Nguồn lực quản trị
 - Tuổi thọ máy chủ thấp
 - Công cụ triển khai chuyển thành CDP - Cloudera Data Platform – 10,000 USD/1 máy chủ
 - Khả năng mở rộng
 - Cân bằng tải
- Yêu cầu đặt ra: Chuyển đổi sang kiến trúc thuần đám mây
 - Sử dụng Kubernetes
 - Mô hình vi dịch vụ
 - Linh hoạt, mở rộng, dễ triển khai

1. Tổng quan

- Kiến trúc dữ liệu xác định:
 - Các thành phần chính
 - Mối quan hệ giữa các thành phần
 - Luồng dữ liệu
 - Các nguyên tắc thiết kế
 - Công nghệ sử dụng
- Các kiểu kiến trúc dữ liệu truyền thống:
 - Hồ dữ liệu
 - Kho dữ liệu
- Các kiểu kiến trúc dữ liệu hiện đại:
 - Kiến trúc 2 tầng kết hợp
 - Data Lakehouse



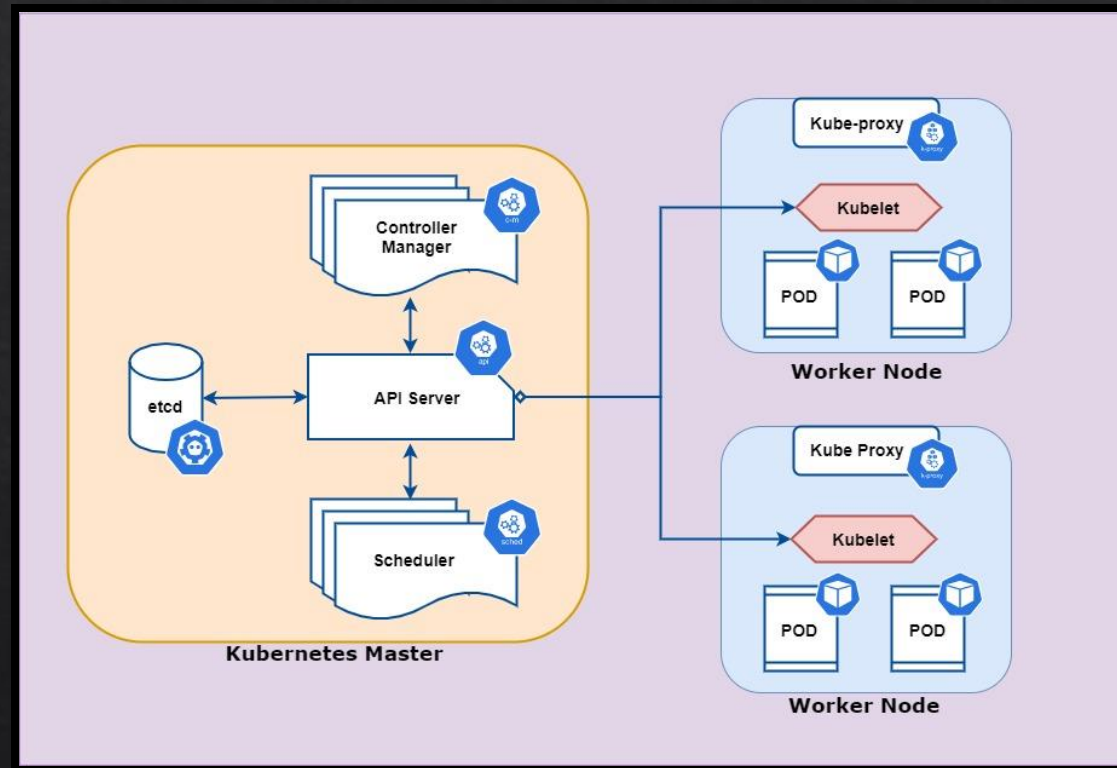
Các thành phần của 1 nền tảng dữ liệu

1. Tổng quan

Tính năng	Kho dữ liệu	Hồ dữ liệu	Kiến trúc kết hợp	Kiến trúc Lakehouse
Kiến trúc	Đơn giản, lưu trữ đơn tầng	Đơn giản, lưu trữ đơn tầng	Cấu trúc 2 tầng phức tạp	Đơn giản, lưu trữ đơn tầng, thống nhất và mở
Kiểu dữ liệu	Phần lớn có cấu trúc và bán cấu trúc	Có cấu trúc, bán cấu trúc và phi cấu trúc	Có cấu trúc, bán cấu trúc và phi cấu trúc	Có cấu trúc, bán cấu trúc và phi cấu trúc
Use cases	Báo cáo BI, MIS, DSS (hệ thống quyết định)	AI/ML, xử lý thời gian thực, BI (hiệu năng giới hạn)	AI/ML, xử lý thời gian thực, BI (hiệu năng giới hạn)	Tất cả các use case - BI, AI, xử lý thời gian thực, ETL
Loại công việc thường xuyên	Theo lô	Theo lô và theo thời gian thực	Theo lô và theo thời gian thực	Theo lô và theo thời gian thực
Tính năng	Hỗ trợ SQL, tính toàn vẹn, dễ dàng cập nhật, xóa	Hỗ trợ cho Spark, không trực tiếp cập nhật/xóa, không hỗ trợ tính toàn vẹn	Hỗ trợ tính toàn vẹn, cập nhật/xóa chỉ cho kho dữ liệu	Hỗ trợ tính toàn vẹn, cập nhật/xóa, và tiến hóa lược đồ
Chất lượng dữ liệu	Chất lượng dữ liệu tốt, ràng buộc lược đồ nghiêm ngặt	Giảm chất lượng dữ liệu, không có ràng buộc lược đồ	Chất lượng tốt cho kho dữ liệu, giảm xuống ở hồ dữ liệu	Chất lượng được cải thiện ở các vùng lưu trữ cao hơn
Kiểm soát truy cập	Được quản trị tốt, kiểm soát truy cập chi tiết	Kiểm soát truy cập mức tệp/đối tượng (object)	Kiểm soát truy cập chi tiết ở kho dữ liệu, ở mức tệp cho hồ dữ liệu	Được quản trị tốt, kiểm soát truy cập chi tiết
Chia sẻ dữ liệu	Người dùng nội bộ hoặc bên ngoài sử dụng cùng một nền tảng cung cấp	Người dùng có thể truy cập trực tiếp các tệp	Mở cho hồ dữ liệu, giới hạn ở kho dữ liệu	Mở, dễ dàng chia sẻ với các người dùng với các nhà cung cấp, nền tảng khác nhau
Hiệu suất	Hiệu suất BI cực tốt	Hiệu suất BI giới hạn	Hiệu suất BI tốt nhất ở kho dữ liệu, giới hạn ở hồ dữ liệu	Hiệu suất BI tốt
Chi phí	Cao do hệ thống độc quyền	Rẻ hơn do không sử dụng hệ thống lưu trữ độc quyền	Chi phí cao do lưu trữ độc quyền của kho dữ liệu, và thêm chi phí đồng bộ giữa hồ dữ liệu và kho dữ liệu	Rẻ hơn do không độc quyền ở hệ thống lưu trữ và tính toán
Rủi ro	Gắn chặt với nhà cung cấp, khó dịch chuyển sang các nhà cung cấp khác	Nếu không có quản trị dữ liệu thì rất dễ trở thành “đầm lầy dữ liệu”	Dữ liệu giữa 2 tầng có thể mất đồng bộ, dữ liệu ở kho dữ liệu gắn chặt với nhà cung cấp	Đây là kiến trúc tương đối mới

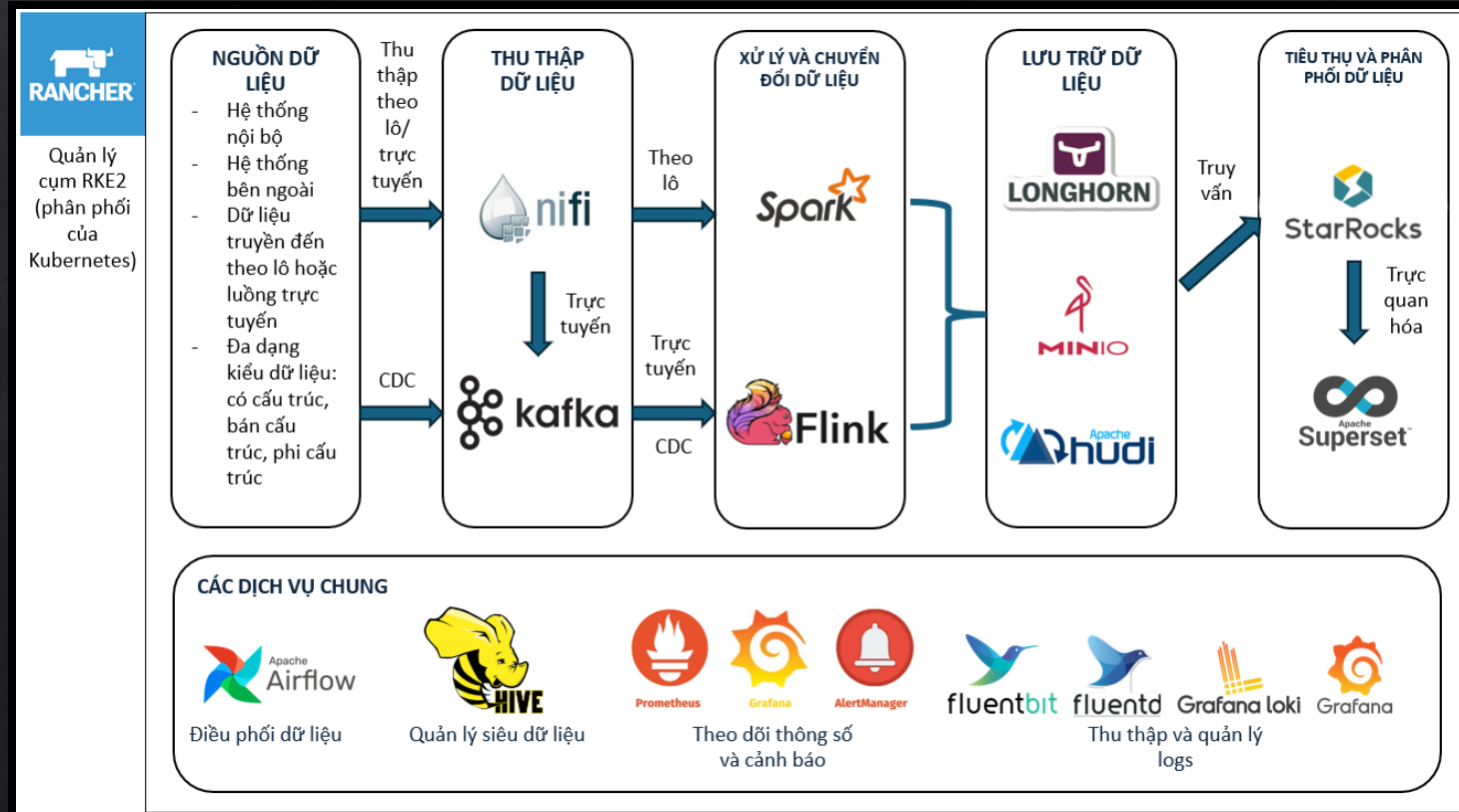
2. Công nghệ sử dụng

- Phương pháp triển khai hệ thống thuần đám mây - Kubernetes



2. Công nghệ sử dụng

- Các công nghệ sử dụng cho Data Lakehouse



3. Thực nghiệm

- Mô tả dữ liệu thực nghiệm:
 - Bộ dữ liệu TPC-H: Dung lượng 10GB

Bảng	Số lượng bản ghi
customer	1,500,000
lineitem	59,986,052
nation	25
orders	15,000,000
part	2,000,000
partsupp	8,000,000
region	5
supplier	100,000

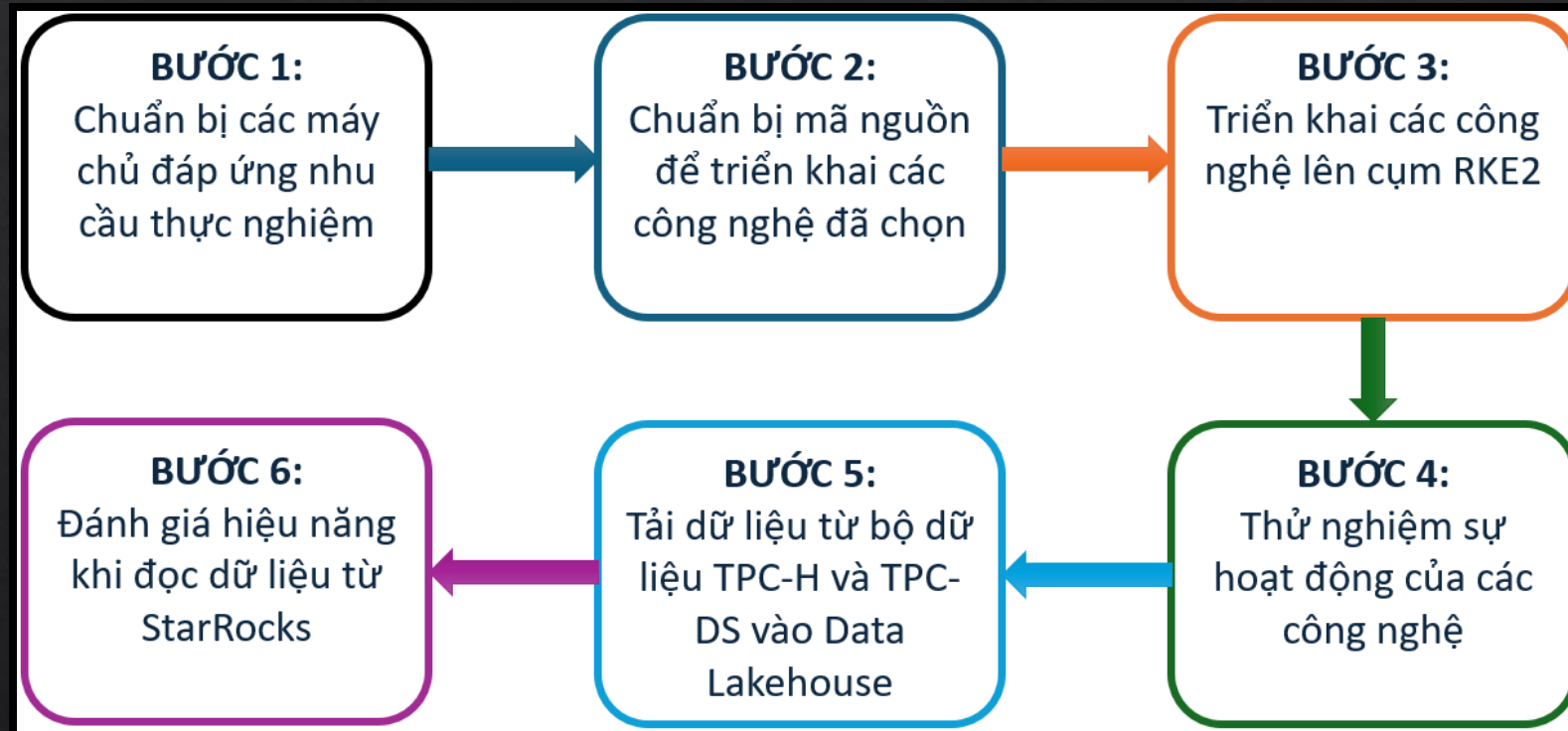
3. Thực nghiệm

- Mô tả dữ liệu thực nghiệm:
 - Bộ dữ liệu TPC-DS: Dung lượng 10GB

Bảng	Số lượng bản ghi
call_center	24
catalog_page	12,000
catalog_returns	1,439,749
catalog_sales	14,401,261
customer_address	250,000
customer_demographics	1,920,800
customer	500,000
date_dim	73,049
household_demographics	7,200
income_band	20
inventory	133,110,000
item	102,000
promotion	500
reason	45
ship_mode	20
store	102
store_returns	2,875,432
store_sales	28,800,991
time_dim	86,400
warehouse	10
web_page	200
web_returns	719,217
web_sales	7,197,566
web_site	42

3. Thực nghiệm

- Quy trình thực nghiệm














3. Thực nghiệm

- Môi trường thực nghiệm: CloudFly

Máy chủ làm Worker chạy các thành phần trên cụm	<ul style="list-style-type: none">- Số lượng: 7- Khu vực: Hồ Chí Minh- CPU: 8 vCPU Intel Xeon V4 xung nhịp 2.6GHz- RAM: 16 GB- Ổ cứng: Enterprise SSD 50 GB- Hệ điều hành: Ubuntu-22.04_64bit- Public IP / Private IP: 103.82.38.174 / 192.168.1.198 103.82.38.104 / 192.168.1.109 103.82.38.64 / 192.168.1.162 103.82.135.105 / 192.168.1.46 103.82.135.68 / 192.168.1.193 103.82.39.191 / 192.168.1.103 103.82.135.168 / 192.168.1.92	Máy chủ chạy Rancher quản lý cụm RKE2	<ul style="list-style-type: none">- Số lượng: 1- Khu vực: Hồ Chí Minh- CPU: 2 vCPU Intel Xeon V4 xung nhịp 2.6GHz- RAM: 4 GB- Ổ cứng: Enterprise SSD 20 GB- Hệ điều hành: Ubuntu-22.04_64bit- Public IP: 103.82.36.206
		Máy chủ làm Control Plane điều hành cụm RKE2	<ul style="list-style-type: none">- Số lượng: 3- Khu vực: Hồ Chí Minh- CPU: 2 vCPU Intel Xeon V4 xung nhịp 2.6GHz- RAM: 4 GB- Ổ cứng: Enterprise SSD 20 GB- Hệ điều hành: Ubuntu-22.04_64bit- Public IP / Private IP: 103.82.37.215 / 192.168.1.97 103.82.37.68 / 192.168.1.160 103.82.132.70 / 192.168.1.35

3. Thực nghiệm

- Kết quả thực nghiệm
 - Bước 1: Chuẩn bị các máy chủ đáp ứng nhu cầu thực nghiệm

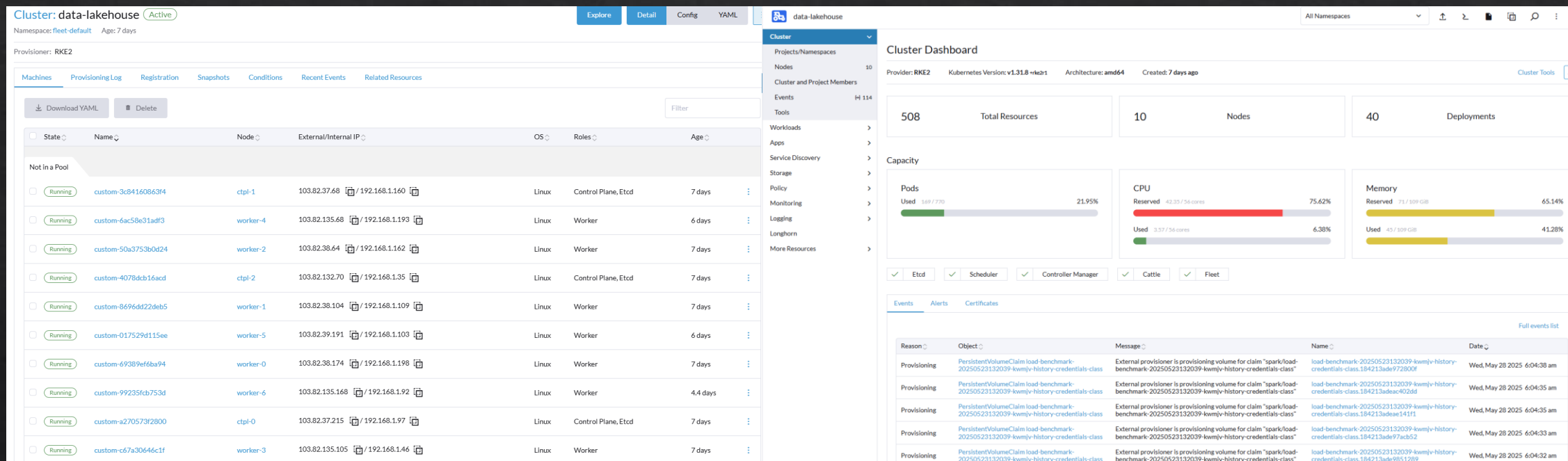
STT	Tên máy chủ	Thời gian	Vị trí	Trạng thái	Tự động giờ
1	 worker-6 Standard hcm-ss-8-16-50 - 103.82.135.168	18:53 23-05-2025	Hồ Chí Minh	ACTIVE	<input checked="" type="checkbox"/>
2	 worker-5 Standard hcm-ss-8-16-50 - 103.82.39.191	08:42 21-05-2025	Hồ Chí Minh	ACTIVE	<input checked="" type="checkbox"/>
3	 worker-4 Standard hcm-ss-8-16-50 - 103.82.135.68	08:08 21-05-2025	Hồ Chí Minh	ACTIVE	<input checked="" type="checkbox"/>
4	 worker-3 Standard hcm-ss-8-16-50 - 103.82.135.105	19:37 20-05-2025	Hồ Chí Minh	ACTIVE	<input checked="" type="checkbox"/>
5	 worker-2 Standard hcm-ss-8-16-50 - 103.82.38.64	09:01 20-05-2025	Hồ Chí Minh	ACTIVE	<input checked="" type="checkbox"/>
6	 worker-1 Standard hcm-ss-8-16-50 - 103.82.38.104	09:01 20-05-2025	Hồ Chí Minh	ACTIVE	<input checked="" type="checkbox"/>
7	 worker-0 Standard hcm-ss-8-16-50 - 103.82.38.174	09:01 20-05-2025	Hồ Chí Minh	ACTIVE	<input checked="" type="checkbox"/>
8	 ctpi-2 Standard hcm-ss-2-4-20 - 103.82.132.70	08:33 20-05-2025	Hồ Chí Minh	ACTIVE	<input checked="" type="checkbox"/>
9	 ctpi-1 Standard hcm-ss-2-4-20 - 103.82.37.68	08:10 20-05-2025	Hồ Chí Minh	ACTIVE	<input checked="" type="checkbox"/>
10	 ctpi-0 Standard hcm-ss-2-4-20 - 103.82.37.215	08:09 20-05-2025	Hồ Chí Minh	ACTIVE	<input checked="" type="checkbox"/>
11	 rancher-server Standard hcm-ss-2-4-20 - 103.82.36.206	13:56 17-05-2025	Hồ Chí Minh	ACTIVE	<input checked="" type="checkbox"/>

3. Thực nghiệm

- Kết quả thực nghiệm
 - Bước 2: Chuẩn bị mã nguồn để triển khai các công nghệ đã chọn:
 - Chuẩn bị các helm chart cần sử dụng cho các công nghệ
 - Cấu hình lại các file values.yaml để cấu hình cho từng công nghệ
 - Viết các file cấu hình (file yaml) mới để áp dụng triển khai công nghệ

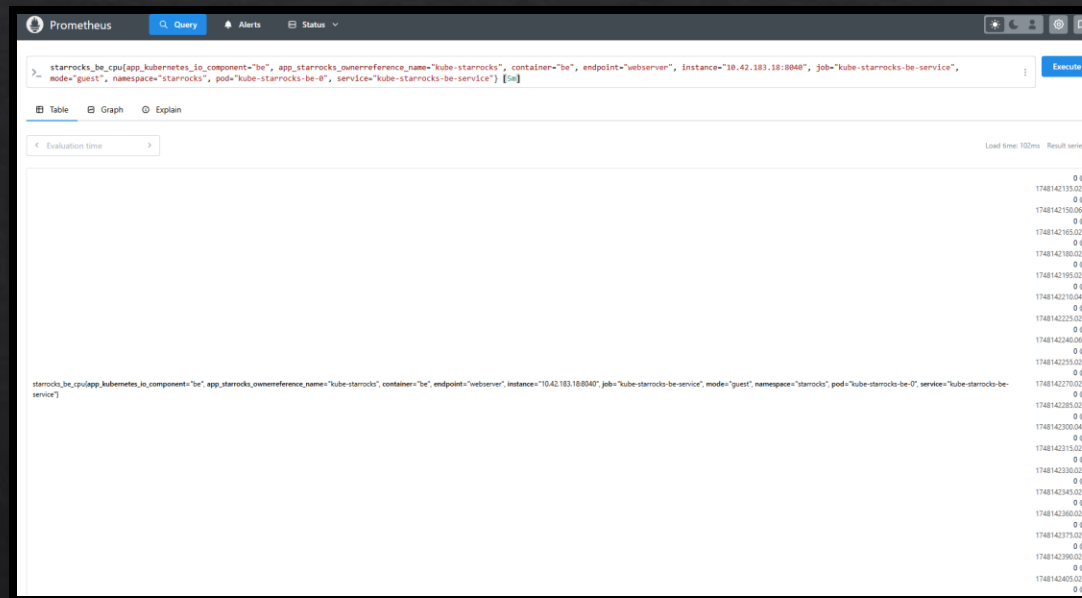
3. Thực nghiệm

- Kết quả thực nghiệm
 - Bước 3: Triển khai các công nghệ lên cụm RKE2

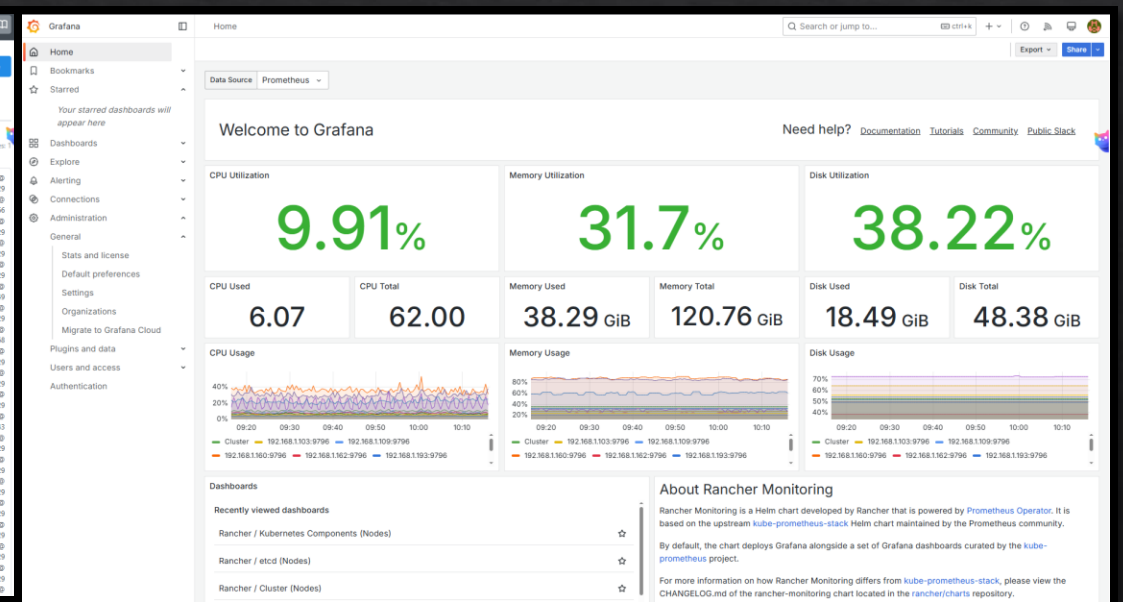


3. Thực nghiệm

- Kết quả thực nghiệm
 - Bước 4: Thử nghiệm sự hoạt động của các công nghệ



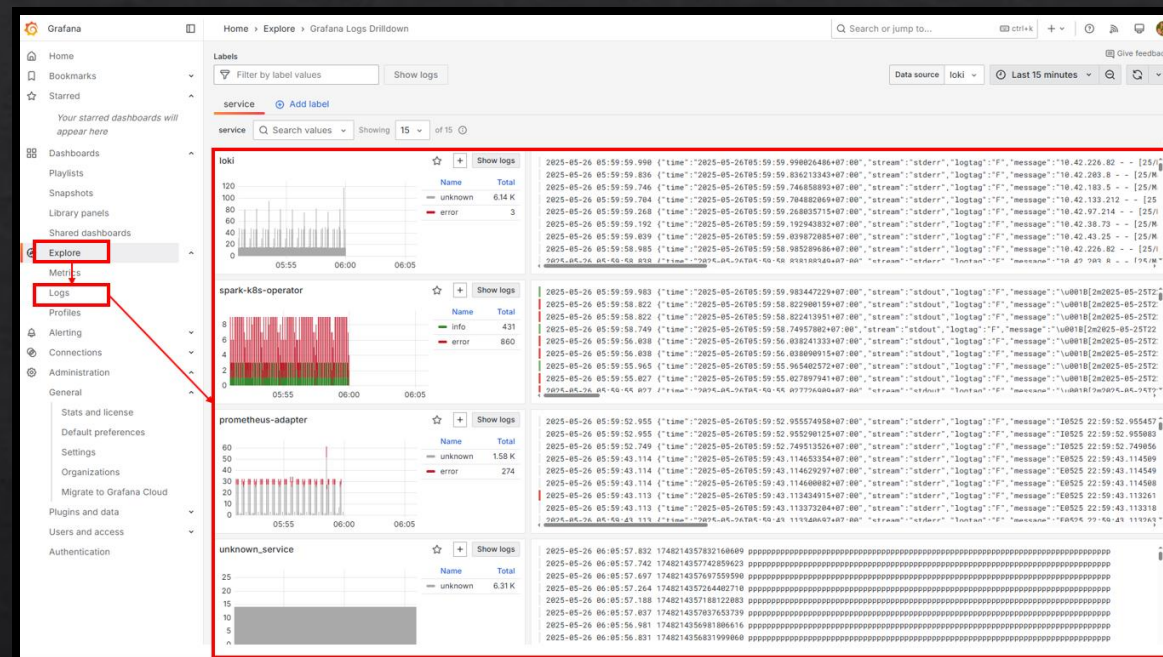
Truy vấn trên Prometheus



Trực quan hóa trên Grafana

3. Thực nghiệm

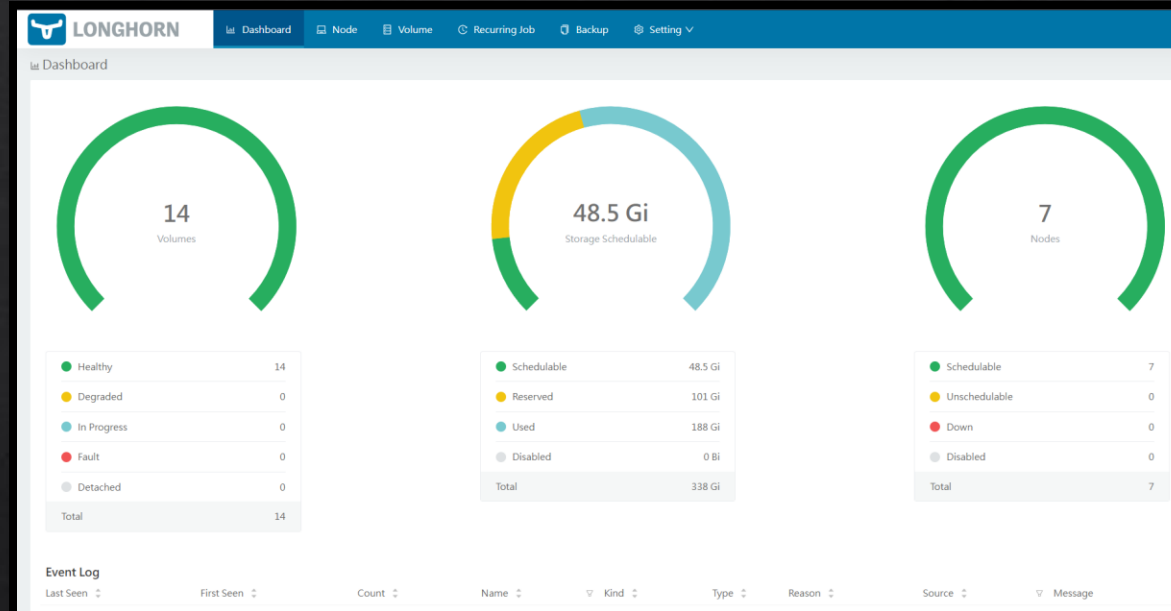
- Kết quả thực nghiệm
 - Bước 4: Thử nghiệm sự hoạt động của các công nghệ



Dữ liệu log trên Loki - Grafana

3. Thực nghiệm

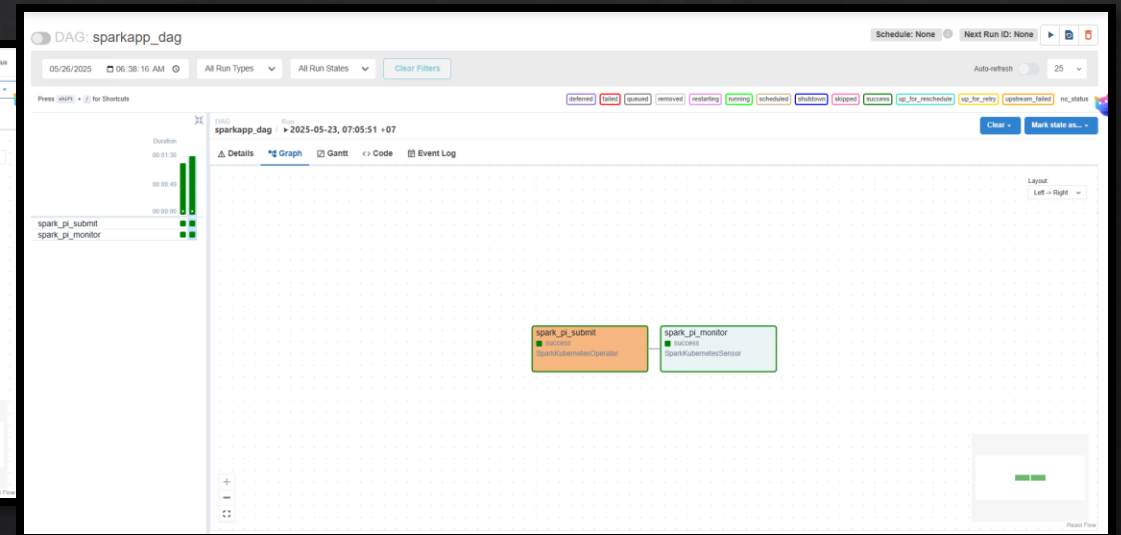
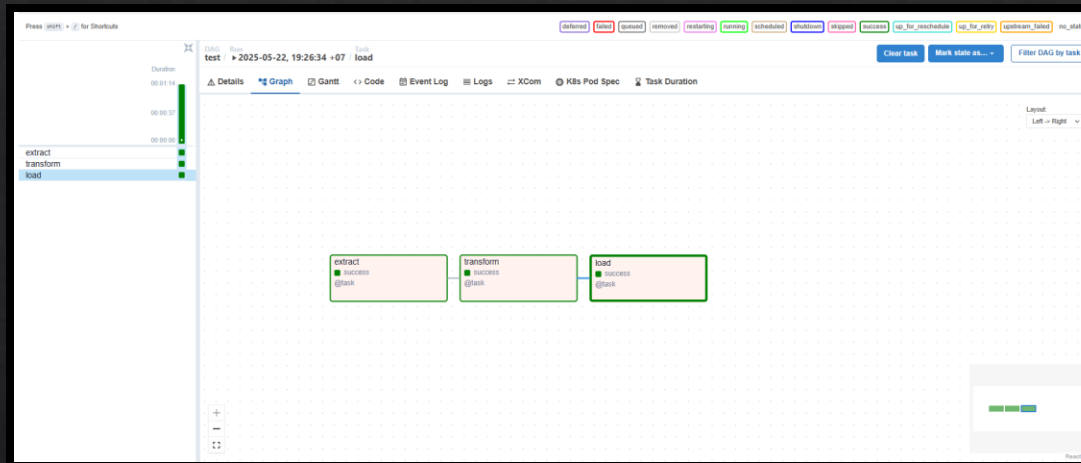
- Kết quả thực nghiệm
 - Bước 4: Thử nghiệm sự hoạt động của các công nghệ



Lưu trữ dài hạn trên Longhorn

3. Thực nghiệm

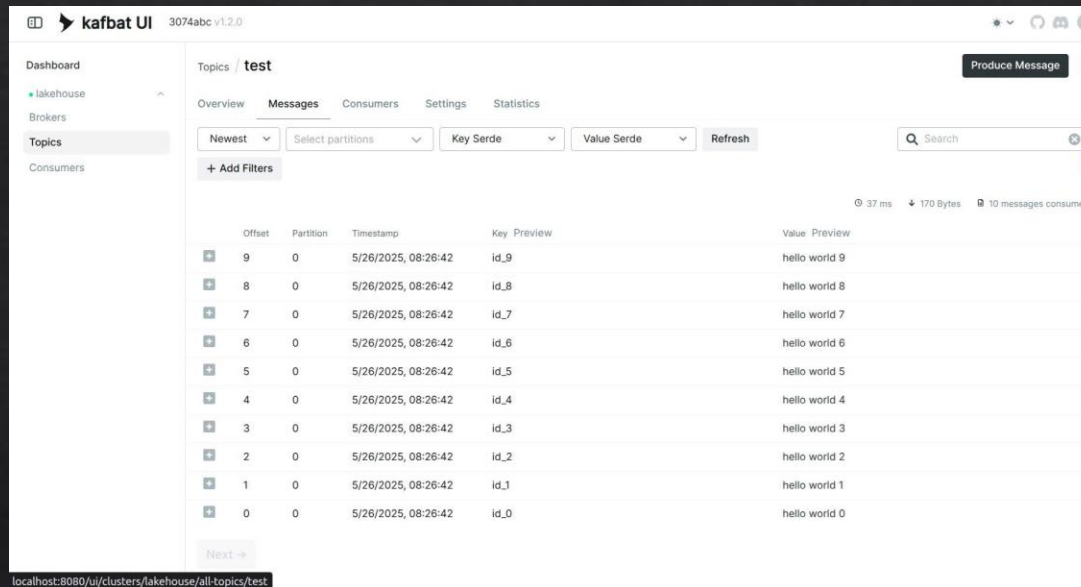
- Kết quả thực nghiệm
 - Bước 4: Thử nghiệm sự hoạt động của các công nghệ



Kết quả chạy thử airflow

3. Thực nghiệm

- Kết quả thực nghiệm
 - Bước 4: Thử nghiệm sự hoạt động của các công nghệ



The screenshot shows the 'kafbat UI' interface for a topic named 'test'. The 'Messages' tab is selected, displaying a table of message details. The table includes columns for Offset, Partition, Timestamp, Key, and Value. The messages are ordered by offset from 0 to 9, with keys 'id_0' through 'id_9' and values 'hello world 0' through 'hello world 9'. The interface also shows a 'Produce Message' button and a search bar.

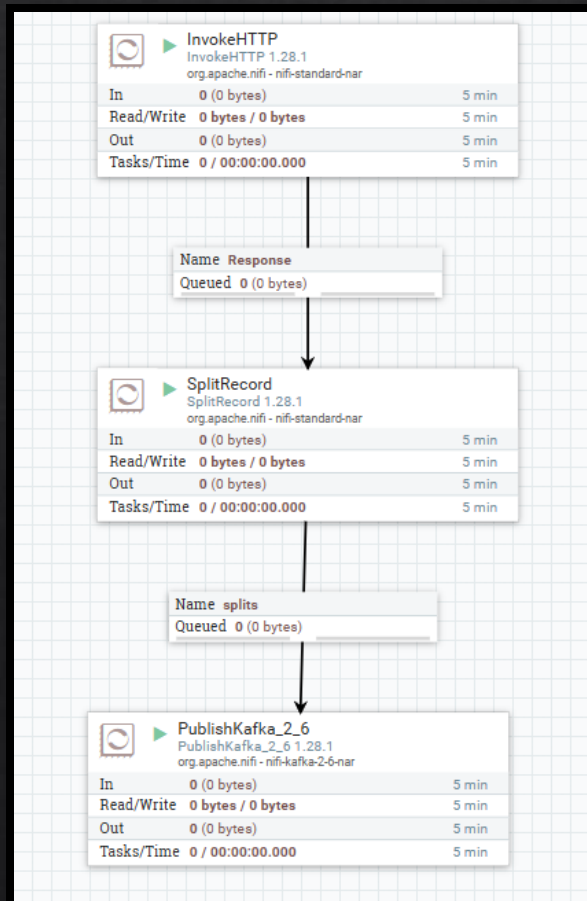
Offset	Partition	Timestamp	Key	Value
9	0	5/26/2025, 08:26:42	id_9	hello world 9
8	0	5/26/2025, 08:26:42	id_8	hello world 8
7	0	5/26/2025, 08:26:42	id_7	hello world 7
6	0	5/26/2025, 08:26:42	id_6	hello world 6
5	0	5/26/2025, 08:26:42	id_5	hello world 5
4	0	5/26/2025, 08:26:42	id_4	hello world 4
3	0	5/26/2025, 08:26:42	id_3	hello world 3
2	0	5/26/2025, 08:26:42	id_2	hello world 2
1	0	5/26/2025, 08:26:42	id_1	hello world 1
0	0	5/26/2025, 08:26:42	id_0	hello world 0

```
ain] INFO com.trieubui1012.kafka.ProducerDemo - Key: id_0, Value: hello world 0
ain] INFO com.trieubui1012.kafka.ProducerDemo - Partition: 0, Offset:0
ain] INFO com.trieubui1012.kafka.ProducerDemo - Key: id_1, Value: hello world 1
ain] INFO com.trieubui1012.kafka.ProducerDemo - Partition: 0, Offset:1
ain] INFO com.trieubui1012.kafka.ProducerDemo - Key: id_2, Value: hello world 2
ain] INFO com.trieubui1012.kafka.ProducerDemo - Partition: 0, Offset:2
ain] INFO com.trieubui1012.kafka.ProducerDemo - Key: id_3, Value: hello world 3
ain] INFO com.trieubui1012.kafka.ProducerDemo - Partition: 0, Offset:3
ain] INFO com.trieubui1012.kafka.ProducerDemo - Key: id_4, Value: hello world 4
ain] INFO com.trieubui1012.kafka.ProducerDemo - Partition: 0, Offset:4
ain] INFO com.trieubui1012.kafka.ProducerDemo - Key: id_5, Value: hello world 5
ain] INFO com.trieubui1012.kafka.ProducerDemo - Partition: 0, Offset:5
ain] INFO com.trieubui1012.kafka.ProducerDemo - Key: id_6, Value: hello world 6
ain] INFO com.trieubui1012.kafka.ProducerDemo - Partition: 0, Offset:6
ain] INFO com.trieubui1012.kafka.ProducerDemo - Key: id_7, Value: hello world 7
ain] INFO com.trieubui1012.kafka.ProducerDemo - Partition: 0, Offset:7
ain] INFO com.trieubui1012.kafka.ProducerDemo - Key: id_8, Value: hello world 8
ain] INFO com.trieubui1012.kafka.ProducerDemo - Partition: 0, Offset:8
ain] INFO com.trieubui1012.kafka.ProducerDemo - Key: id_9, Value: hello world 9
ain] INFO com.trieubui1012.kafka.ProducerDemo - Partition: 0, Offset:9
```

Kết quả chạy thử Kafka

3. Thực nghiệm

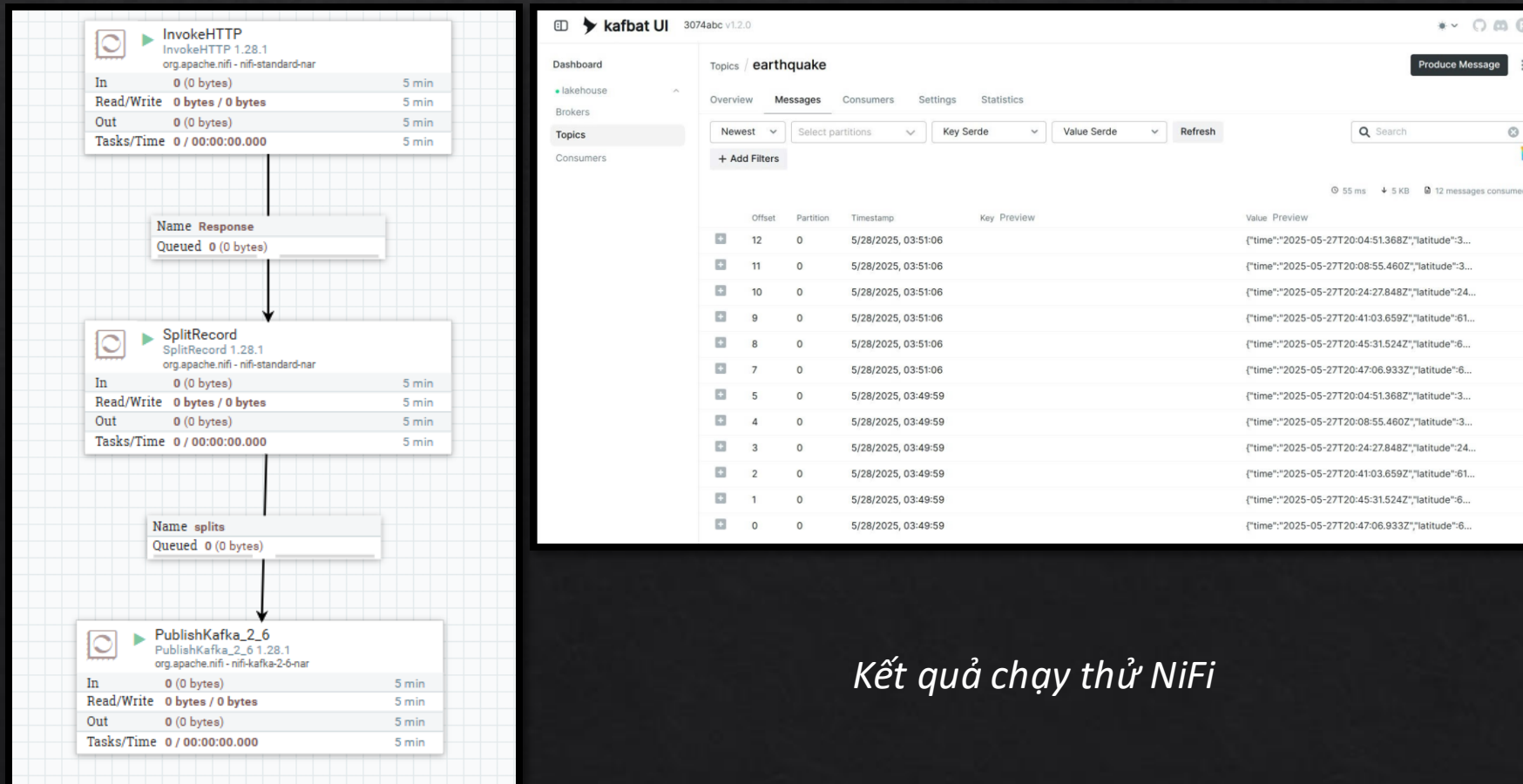
- Kết quả thực nghiệm
 - Bước 4: Thử nghiệm sự hoạt động của các công nghệ



Kết quả chạy thử NiFi

3. Thực nghiệm

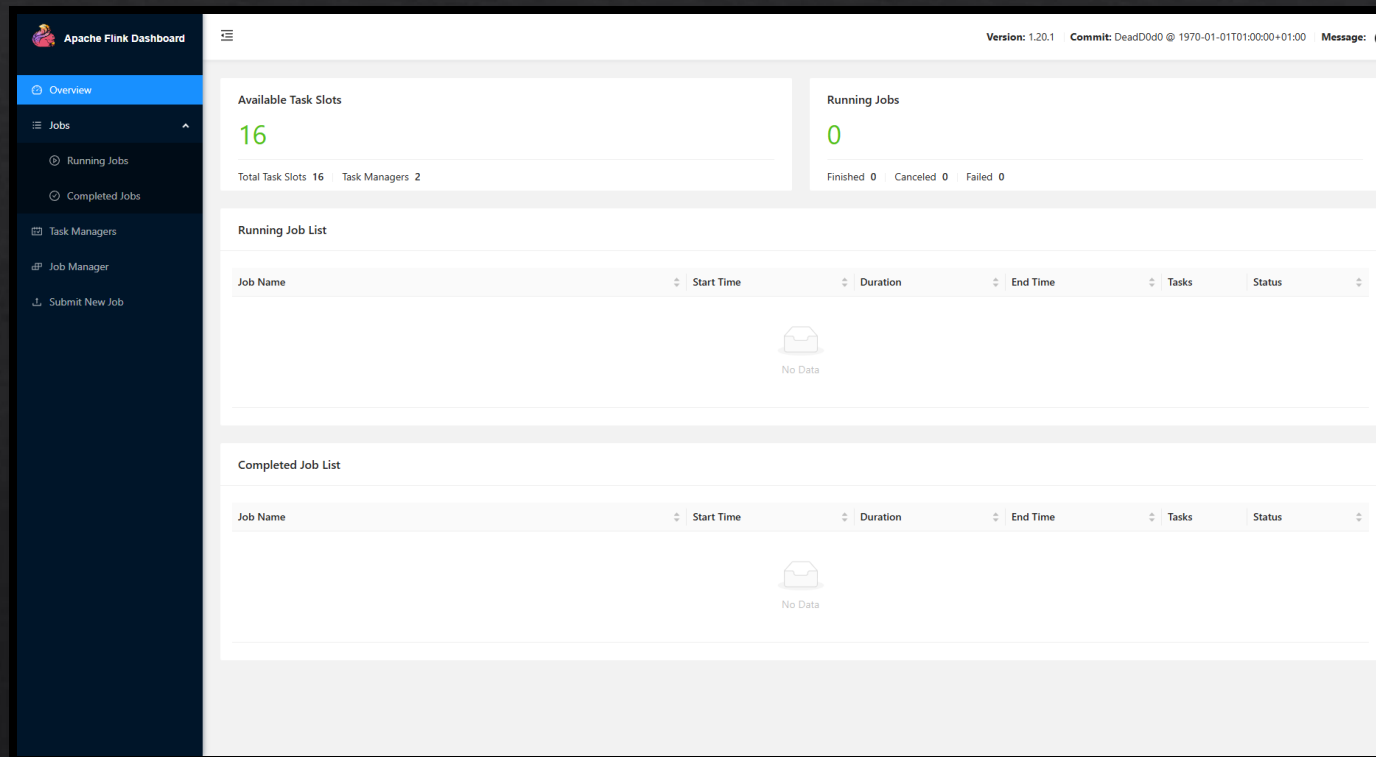
- Kết quả thực nghiệm
 - Bước 4: Thử nghiệm sự hoạt động của các công nghệ



Kết quả chạy thử NiFi

3. Thực nghiệm

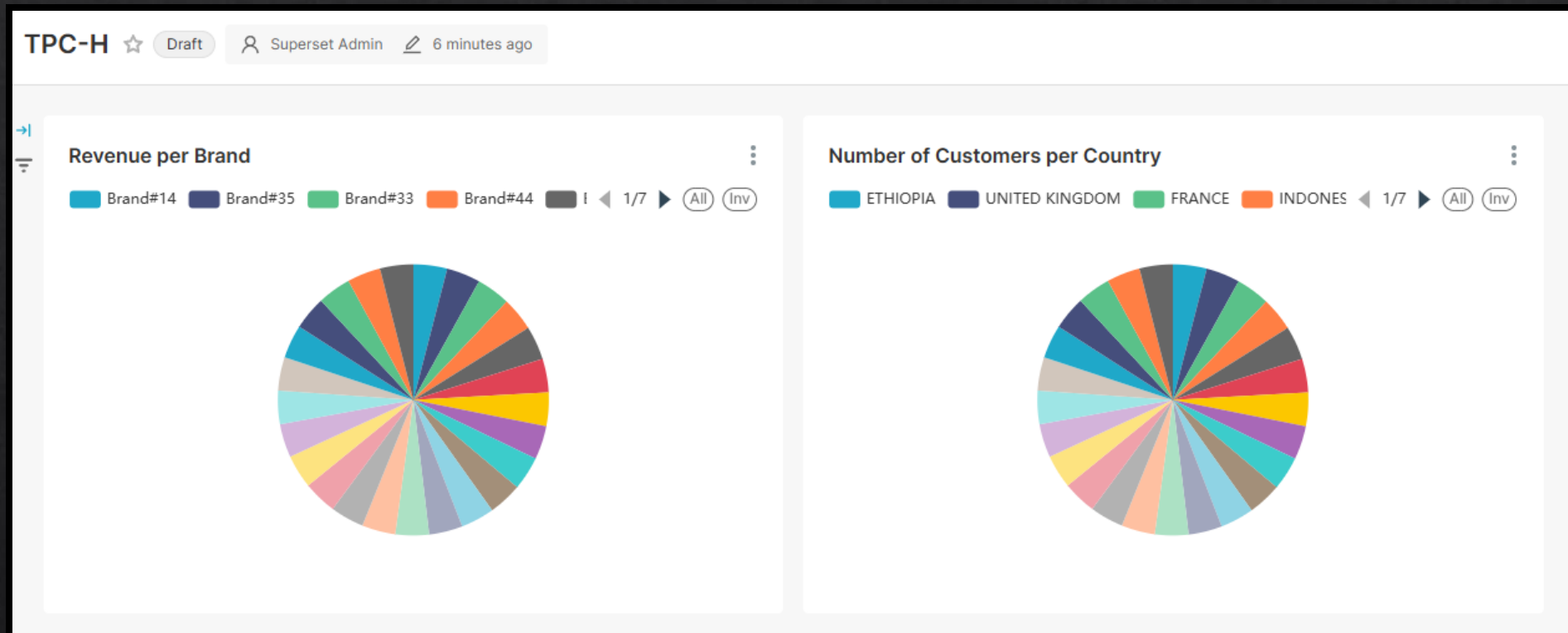
- Kết quả thực nghiệm
 - Bước 4: Thử nghiệm sự hoạt động của các công nghệ



Kết quả chạy thử Flink

3. Thực nghiệm

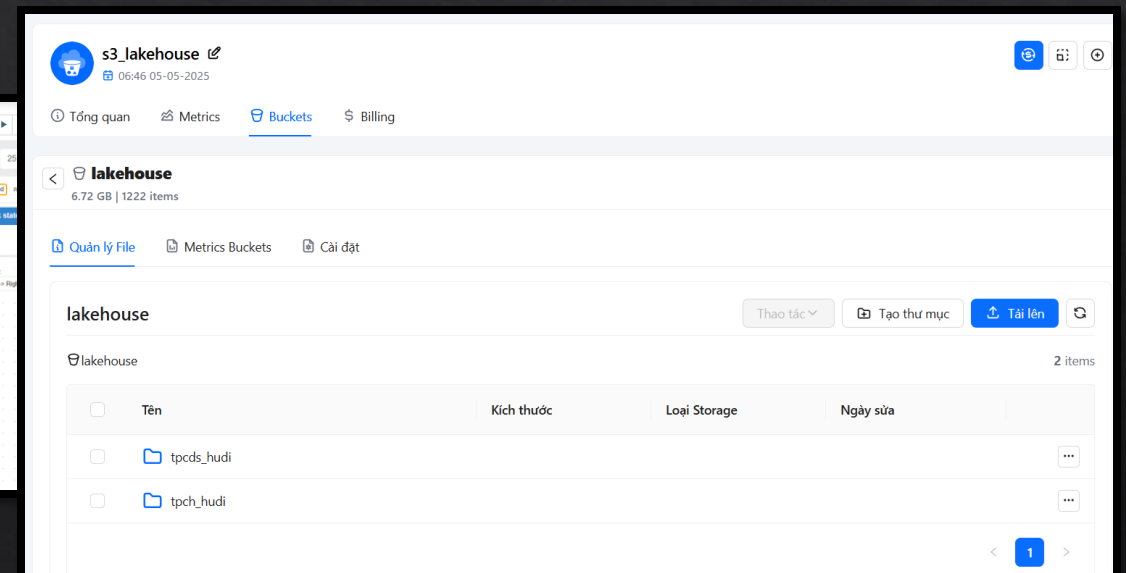
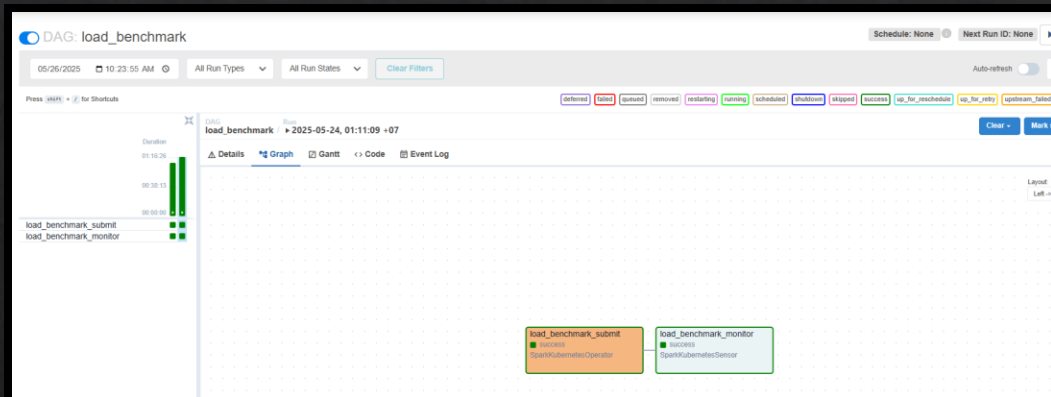
- Kết quả thực nghiệm
 - Bước 4: Thử nghiệm sự hoạt động của các công nghệ



Kết quả chạy thử Superset

3. Thực nghiệm

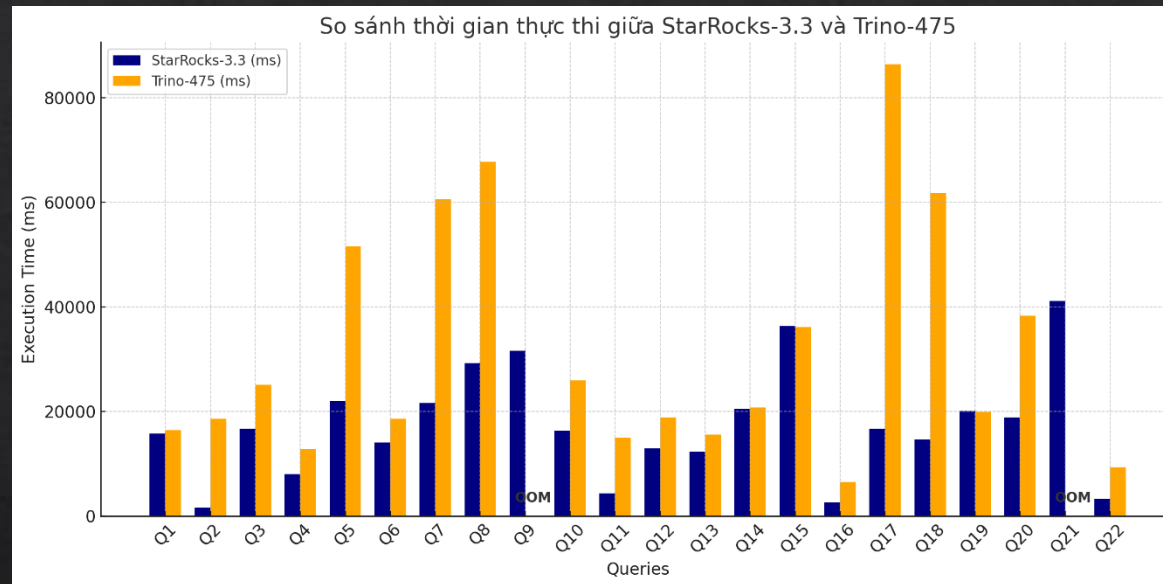
- Kết quả thực nghiệm
 - Bước 5: Tải dữ liệu từ bộ dữ liệu TPC-H và TPC-DS vào Data Lakehouse



Ghi dữ liệu thành công

3. Thực nghiệm

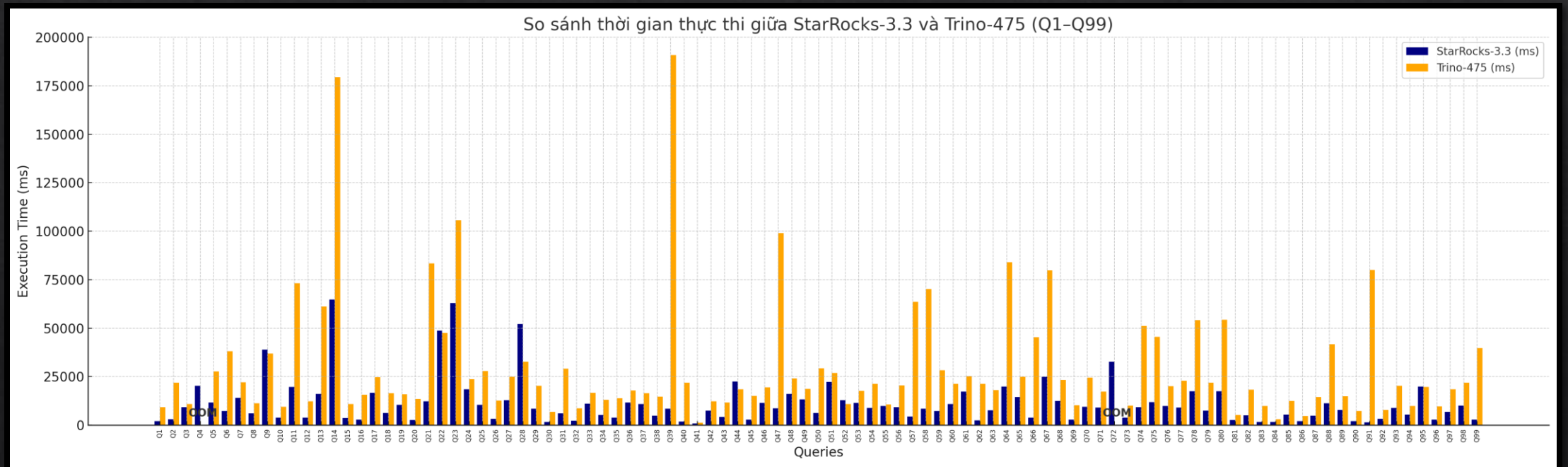
- Kết quả thực nghiệm
 - Bước 6: Đánh giá hiệu năng khi đọc dữ liệu từ StarRocks



Bộ dữ liệu TPC-H

3. Thực nghiệm

- Kết quả thực nghiệm
 - Bước 6: Đánh giá hiệu năng khi đọc dữ liệu từ StarRocks



Bộ dữ liệu TPC-DS

3. Thực nghiệm

- Đánh giá
 - Đánh giá chung:
 - Tất cả các công nghệ đều được triển khai thành công và hoạt động tốt trên hệ thống. Đây sẽ là tiền đề quan trọng để đưa hệ thống vào triển khai thực tế trong tương lai.
 - Ngoài ra đánh giá hiệu năng đọc dữ liệu từ Hudi trên StarRocks và Trino (1 công cụ truy vấn nổi tiếng hiện nay) cho thấy StarRocks vượt trội hơn rất nhiều so với Trino.

3. Thực nghiệm

- Đánh giá:
 - Đánh giá khả năng đọc trên Data Lakehouse: Tập dữ liệu TPC-H:
 - Số truy vấn nhanh hơn: StarRocks nhanh hơn Trino ở 20/22 truy vấn (không xét hai trường hợp Trino OOM – tràn bộ nhớ: Q9 và Q21). Chỉ có Q15 (36,351 ms vs 36,100 ms) và Q19 (20,148 ms vs 19,930 ms) Trino nhỉnh hơn, nhưng khoảng chênh rất nhỏ (0.7% và 1.1%).
 - Ổn định bộ nhớ: Trino bị tràn bộ nhớ trên Q9 và Q21, trong khi StarRocks hoàn thành cả 22 truy vấn mà không gặp lỗi bộ nhớ.
 - Tổng thời gian thực thi: StarRocks nhanh hơn Trino khoảng 39% tổng thời gian.
 - Những chênh lệch lớn nhất: Q17 (16.7 s và 86.4 s - Trino chậm hơn xấp xỉ 5 lần), Q8 (29.3 s và 67.8 s - Trino chậm hơn xấp xỉ 2.3 lần), Q7 (21.6 s và 60.6 s – Trino chậm hơn xấp xỉ 2.8 lần)

3. Thực nghiệm

- Đánh giá:
 - Đánh giá khả năng đọc trên Data Lakehouse: Tập dữ liệu TPC-DS:
 - Số truy vấn: StarRocks nhanh hơn Trino ở 93/99 truy vấn (~94%). Trino nhanh hơn StarRocks chỉ ở 6/99 truy vấn (Q09, Q22, Q28, Q44, Q52, Q95), chiếm khoảng 6%.
 - Tổng thời gian: StarRocks xử lý xong toàn bộ nhanh hơn Trino khoảng 61.5% tổng thời gian.
 - Ổn định bộ nhớ: Trino bị tràn bộ nhớ trên 2 truy vấn (Q04, Q72), trong khi StarRocks hoàn thành cả 99 truy vấn mà không lỗi bộ nhớ.
 - Những chênh lệch lớn nhất: Q91 (Trino chậm hơn khoảng 60 lần), Q39 (Trino chậm hơn khoảng 23 lần), Q57 (Trino chậm hơn khoảng 14.5 lần), Q99 (Trino chậm hơn khoảng 14.4 lần), Q40 (Trino chậm hơn khoảng 12 lần)

CẢM ƠN QUÝ THẦY CÔ VÀ CÁC
BẠN ĐÃ CHÚ Ý LẮNG NGHE