

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỆ THỐNG THÔNG TIN



BÁO CÁO SEMINAR CHƯƠNG 5

Môn học: Dữ liệu lớn

Đề tài: **APACHE DORIS**

Mã lớp: IS405.O11.HTCL

Giảng viên hướng dẫn: ThS. Nguyễn Hồ Duy Tri

Sinh viên thực hiện:

Trần Phương Thảo 20521938

Phạm Thụy Ý Vy 20522183

Lưu Yên Vy 20522180

TP. Hồ Chí Minh, 2023

MỤC LỤC

LỜI CẢM ƠN	4
NHẬN XÉT CỦA GIẢNG VIÊN.....	5
CHƯƠNG 1:GIỚI THIỆU CHUNG	6
1.1 Tổng quan về Apache Doris	6
1.2 Lịch sử hình thành và phát triển	6
1.3 Ứng dụng trong các doanh nghiệp	6
1.4 Các đặc điểm chính	7
1.4.1 Kiến trúc.....	7
1.4.2 Công cụ lưu trữ.....	8
1.4.3 Mô hình lưu trữ	9
1.4.4 Công cụ truy vấn	10
1.4.5 Trình tối ưu hóa truy vấn	12
CHƯƠNG 2:ƯU ĐIỂM VÀ NHƯỢC ĐIỂM	13
2.1 Ưu điểm	13
2.2 Nhược điểm.....	13
2.3 So sánh với các sản phẩm cùng loại	14
2.3.1 Apache Druid	14
2.3.2 Apache Kylin.....	15
2.3.3 ClickHouse	16
CHƯƠNG 3:CÁC TRƯỜNG HỢP CỤ THỂ ÁP DỤNG APACHE DORIS	17
3.1 Phân tích báo cáo	17
3.2 Truy vấn Ad-hoc	18
3.3 Xây dựng kho dữ liệu đồng nhất	18

3.4	Tăng tốc độ truy vấn Data Lake.....	18
CHƯƠNG 4:HƯỚNG DẪN CÀI ĐẶT VÀ CẤU HÌNH.....		19
4.1	Cài đặt Apache Doris	19
4.2	Cấu hình Apache Doris.....	19
4.2.1	Cấu hình Frontend (FE)	19
4.2.2	Cấu hình Backend (BE)	23
4.3	Minh họa truy vấn dữ liệu.....	26
4.3.1	Mô tả dữ liệu	26
4.3.2	Truy vấn dữ liệu	29
NGUỒN THAM KHẢO		31

LỜI CẢM ƠN

Trước hết, chúng em xin gửi tới các thầy, cô khoa Hệ thống thông tin, thuộc trường Đại học Công nghệ thông tin – Đại học quốc gia TP. HCM lời cảm ơn vì đã tận tâm truyền đạt kiến thức, hướng dẫn, đặt nền tảng cơ bản cho chúng em có thể thực hiện đồ án này.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành đến **Thầy Nguyễn Hồ Duy Tri**

Để đồ án này được đạt kết quả tốt như hiện nay, chúng em đã nhận được rất nhiều sự hỗ trợ và hướng dẫn từ thầy. Mặc dù đã nỗ lực cố gắng hết sức nhưng do kiến thức còn nhiều mặt hạn chế, nên trong quá trình thực hiện không tránh khỏi những thiếu sót. Kính mong nhận được sự góp ý và giúp đỡ từ quý thầy cô để chúng em có thể hoàn thiện đồ án một cách trọn vẹn nhất.

Chúng em xin chân thành cảm ơn!

Nhóm sinh viên thực hiện:

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

CHƯƠNG 1: GIỚI THIỆU CHUNG

1.1 Tổng quan về Apache Doris

Apache Doris là một cơ sở dữ liệu phân tích thời gian thực, hiệu suất cao dựa trên kiến trúc MPP (Massively Parallel Processing), được biết đến với tốc độ xử lý nhanh chóng và dễ sử dụng. Với thời gian phản hồi dưới một giây để trả về kết quả truy vấn đối với lượng dữ liệu lớn, Apache Doris không chỉ hỗ trợ truy vấn đồng thời mà còn cả các phân tích phức tạp với thông lượng cao. Dựa trên điều này, Apache Doris có thể đáp ứng tốt các tính năng như: phân tích báo cáo, truy vấn ad-hoc, xây dựng kho dữ liệu thống nhất, tăng tốc độ truy vấn Data Lake, ...

1.2 Lịch sử hình thành và phát triển

Apache Doris lần đầu tiên ra đời với tên gọi là Palo, được tạo ra nhằm hỗ trợ hoạt động báo cáo về hiệu quả của các chiến dịch quảng cáo của Baidu. Apache Doris chính thức có mã nguồn mở vào năm 2017 và được Baidu đóng góp cho Apache Foundation để phát triển vào tháng 7 năm 2018. Hiện nay, cộng đồng Apache Doris đã thu hút hơn 400 cộng tác viên từ hàng trăm công ty thuộc các ngành khác nhau và số lượng cộng tác viên tích cực là gần 100 người mỗi tháng.

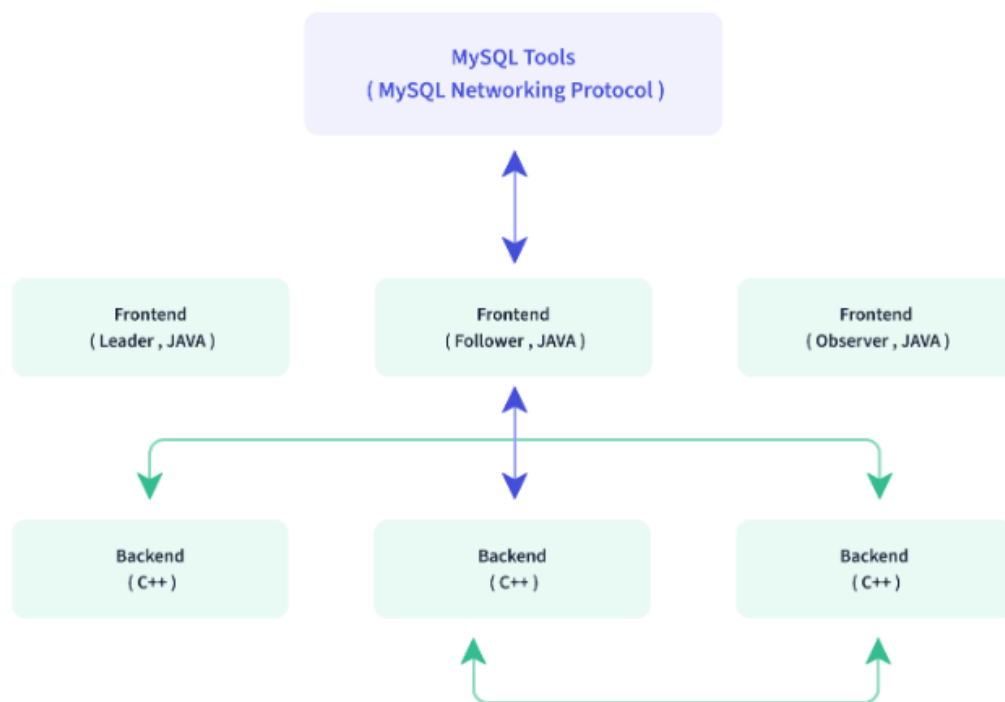
1.3 Ứng dụng trong các doanh nghiệp

Apache Doris hiện có cơ sở người dùng rộng rãi ở Trung Quốc và trên toàn thế giới, và tính đến ngày hôm nay, Apache Doris được sử dụng rộng rãi trong môi trường sản xuất tại hơn 2000 công ty trên toàn thế giới. Hơn 80% trong số 50 công ty Internet hàng đầu ở Trung Quốc là người dùng lâu dài của Apache Doris bao gồm: Baidu, Meituan, Xiaomi, Jingdong, Bytedance, Tencent, NetEase, Kwai, Weibo... Ngoài ra, Apache Doris cũng được sử dụng rộng rãi

trong một số ngành công nghiệp truyền thống như tài chính, năng lượng, sản xuất và viễn thông.

1.4 Các đặc điểm chính

1.4.1 Kiến trúc



Là một hệ thống phân tán lưu trữ dữ liệu, sử dụng giao thức MySQL và hỗ trợ SQL tiêu chuẩn. Apache Doris có kiến trúc đơn giản, chỉ với 2 loại quy trình:

- *Frontend (FE)*: chịu trách nhiệm chính về yêu cầu truy cập của người dùng, phân tích cú pháp truy vấn và lập kế hoạch, quản lý meta data và các công việc liên quan đến quản lý node.

- *Backend (BE)*: chịu trách nhiệm chính cho việc lưu trữ dữ liệu và thực hiện kế hoạch truy vấn.

Cả 2 loại quy trình đều có thể mở rộng theo chiều ngang, cho phép một cụm Apache Doris duy nhất hỗ trợ đến hàng trăm máy và dung lượng lưu trữ hàng chục petabyte. Ngoài ra, Apache Doris còn đảm bảo tính khả dụng cao của các dịch vụ và độ tin cậy cao của dữ liệu thông qua các giao thức nhất quán. Thiết kế kiến trúc tích hợp cao này làm giảm đáng kể chi phí vận hành và bảo trì của một hệ thống phân tán.

1.4.2 Công cụ lưu trữ

Apache Doris sử dụng lưu trữ dạng cột để mã hóa và nén và đọc dữ liệu theo cột. Điều này cho phép Apache Doris đạt được tỷ lệ nén rất cao, đồng thời giảm số lượng lớn các lần quét dữ liệu không liên quan. Kết quả là, Apache Doris sử dụng hiệu quả hơn tài nguyên IO và CPU.

Apache Doris cũng hỗ trợ cấu trúc chỉ mục tương đối phong phú để giảm việc quét dữ liệu. Cụ thể, Apache Doris hỗ trợ các loại chỉ mục sau:

- *Chỉ mục khóa ghép được sắp xếp*: Có thể chỉ định tối đa ba cột để tạo thành khóa sắp xếp phức hợp. Với chỉ mục này, dữ liệu có thể được lược bớt một cách hiệu quả để hỗ trợ tốt hơn cho các kịch bản báo cáo đồng thời cao.
- *Chỉ mục Z-order*: Sử dụng lập chỉ mục thứ tự Z, bạn có thể chạy các truy vấn phạm vi một cách hiệu quả trên bất kỳ tổ hợp trường nào trong lược đồ của bạn.
- *MIN / MAX indexing*: Lọc hiệu quả các truy vấn tương đương và phạm vi cho các loại số.

- *Bộ lọc Bloom*: rất hiệu quả để lọc tương đương và cắt bớt các cột có số lượng cao.
- *Đảo ngược chỉ mục*: Nó cho phép tìm kiếm nhanh bất kỳ trường nào

Các loại chỉ mục này cho phép Doris cung cấp khả năng truy vấn dữ liệu hiệu quả cho nhiều loại truy vấn khác nhau. Ví dụ: chỉ mục khóa ghép được sắp xếp rất hiệu quả cho các truy vấn tổng hợp, chỉ mục Z-order rất hiệu quả cho các truy vấn phạm vi, và bộ lọc Bloom rất hiệu quả cho các truy vấn tương đương.

Nhìn chung, khả năng lưu trữ và truy vấn dữ liệu hiệu quả của Apache Doris là một trong những ưu điểm chính của hệ thống này.

1.4.3 Mô hình lưu trữ

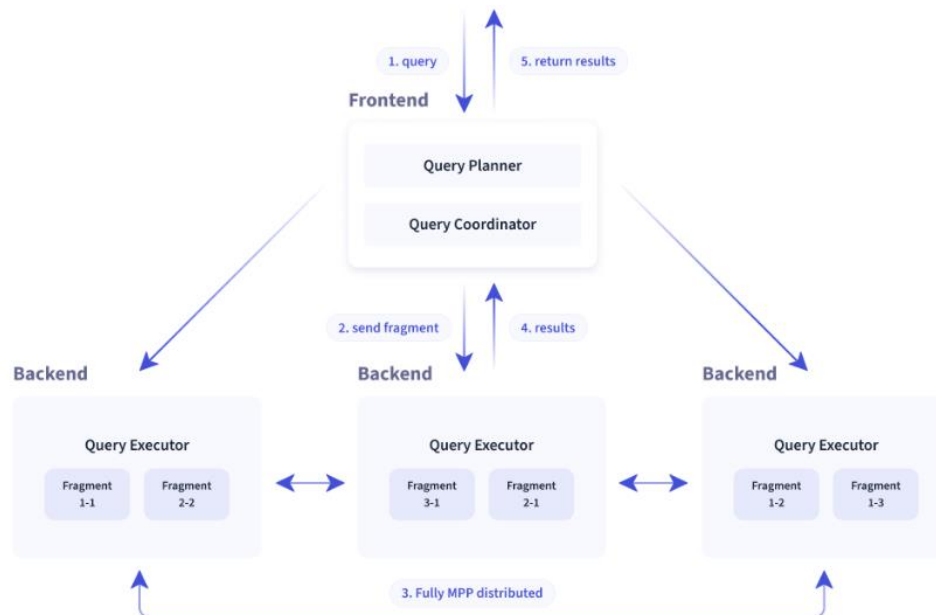
Apache Doris hỗ trợ nhiều mô hình lưu trữ khác nhau, mỗi mô hình có ưu điểm và nhược điểm riêng. Các mô hình lưu trữ này được tối ưu hóa cho các tình huống khác nhau, chẳng hạn như:

- *Mô hình khóa tổng hợp*: Hợp nhất các cột giá trị với các khóa giống nhau, bằng cách tổng hợp trước để cải thiện đáng kể hiệu suất cho các truy vấn tổng hợp.
- *Mô hình khóa duy nhất*: Chìa khóa là duy nhất. Dữ liệu có cùng khóa sẽ được ghi đè để đạt được cập nhật dữ liệu cấp hàng.
- *Mô hình khóa trùng lặp*: Mô hình dữ liệu chi tiết có thể đáp ứng việc lưu trữ chi tiết các bảng dữ liệu.

Ngoài ra, Doris cũng hỗ trợ các chế độ xem cụ thể hóa nhất quán mạnh mẽ. Các chế độ xem cụ thể hóa là các bản sao của các bảng dữ liệu được tạo ra để cải thiện hiệu suất của các truy vấn thường xuyên. Doris tự động

cập nhật và lựa chọn các chế độ xem cụ thể hóa, do đó giảm đáng kể chi phí bảo trì chế độ xem cụ thể hóa.

1.4.4 Công cụ truy vấn



Apache Doris sử dụng kiến trúc MPP. Kiến trúc MPP cho phép Doris thực hiện song song các truy vấn giữa và trong các nút, cũng như hỗ trợ kết hợp trộn phân tán cho nhiều bảng lớn. Điều này giúp Doris có thể đối phó tốt hơn với các truy vấn phức tạp.

Ngoài ra, Apache Doris còn sử dụng công cụ truy vấn được vector hóa và tất cả các cấu trúc bộ nhớ có thể được sắp xếp theo định dạng cột. Điều này giúp giảm đáng kể các lệnh gọi hàm ảo, cải thiện tỷ lệ truy cập bộ nhớ cache và sử dụng hiệu quả các lệnh SIMD. Hiệu suất trong các kịch bản tổng hợp bảng rộng cao hơn 5–10 lần so với các công cụ không vector hóa.

- *SIMD* là một kiểu xử lý song song trong phân loại của Flynn. SIMD mô tả các máy tính có nhiều phần tử xử lý thực hiện cùng một thao tác trên nhiều điểm dữ liệu đồng thời. SIMD đặc biệt có thể áp dụng

cho các tác vụ phổ biến như điều chỉnh độ tương phản trong hình ảnh kỹ thuật số hoặc điều chỉnh âm lượng của âm thanh kỹ thuật số. Hầu hết các thiết kế CPU hiện đại bao gồm các hướng dẫn SIMD để cải thiện hiệu suất sử dụng đa phương tiện.

- *MPP*

- Là một nền tảng tính toán dữ liệu có khả năng xử lý dữ liệu với tốc độ vượt trội. Ở đó, hàng trăm hoặc hàng nghìn hệ thống xử lý cùng hoạt động trên các phần khác nhau của chương trình, mỗi hệ thống có bộ nhớ và vận hành riêng.
- Về cơ bản, MPP sử dụng phân đoạn dữ liệu và việc tạo các khối dữ liệu qua các nút, từ đó xử lý chúng đồng thời cùng lúc. MPP không tiến hành trên các phần cứng thương mại. Thay vào đó, nó dùng các thiết bị đặc thù, có dung lượng bộ nhớ cao. Giao diện kiểu SQL hỗ trợ truy xuất dữ liệu và chúng xử lý dữ liệu nhanh hơn nhờ xử lý trong bộ nhớ.

- Lợi ích của kiến trúc MPP:

- *Sự linh hoạt*: Có thể dễ dàng thêm vào các nodes để mở rộng quy mô.
- *Tối ưu chi phí*: Với MPP, chi phí đầu tư vào phần cứng sẽ được tiết kiệm đáng kể.
- *Đáng tin cậy*: Hệ thống cơ sở dữ liệu này giảm thiểu tỉ lệ xảy ra sai sót.
- *Khả năng mở rộng*: Với MPP, có vô số cách để mở rộng quy mô.

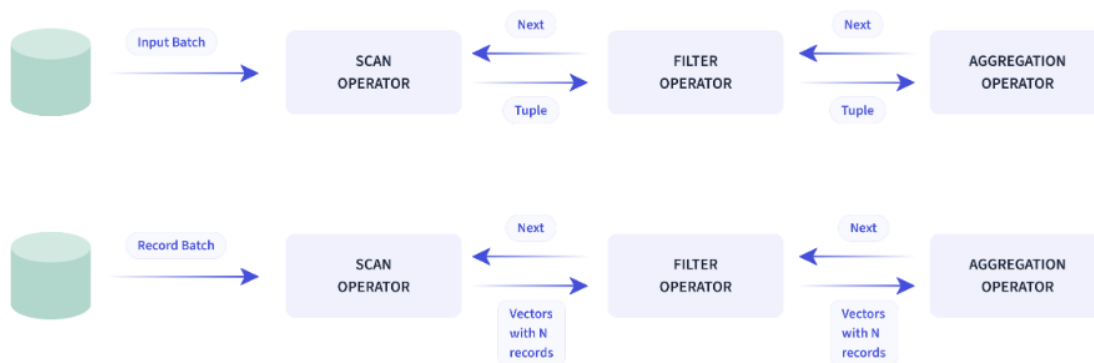
1.4.5 Trình tối ưu hóa truy vấn

Doris sử dụng kết hợp giữa CBO (Cost-Based Optimization) và RBO (Rule-Based Optimization) để tối ưu hóa truy vấn. RBO hỗ trợ các thao tác tối ưu hóa cơ bản như gấp liên tục, viết lại truy vấn con, đẩy xuống vị từ, v.v., trong khi CBO hỗ trợ các thao tác tối ưu hóa phức tạp hơn như tham gia sắp xếp lại.

CBO là một phương pháp tối ưu hóa truy vấn dựa trên chi phí. CBO sử dụng thông tin thống kê về dữ liệu để ước tính chi phí của các kế hoạch truy vấn khác nhau. Sau đó, CBO sẽ chọn kế hoạch có chi phí thấp nhất.

RBO là một phương pháp tối ưu hóa truy vấn dựa trên quy tắc. RBO sử dụng các quy tắc chung để tối ưu hóa truy vấn. Các quy tắc này thường được dựa trên kinh nghiệm hoặc các nghiên cứu đã được thực hiện.

Apache Doris đang tiếp tục tối ưu hóa CBO, tập trung vào việc thu thập và dẫn xuất thông tin thống kê chính xác hơn, dự đoán mô hình chi phí chính xác hơn, v.v. Điều này sẽ giúp CBO tạo ra các kế hoạch truy vấn hiệu quả hơn.



CHƯƠNG 2: ƯU ĐIỂM VÀ NHƯỢC ĐIỂM

2.1 Ưu điểm

- *Hiệu suất cao:* Apache Doris được thiết kế để xử lý các truy vấn OLAP với hiệu suất cao, đặc biệt là trong các trường hợp nơi cần phải xử lý lượng dữ liệu lớn.
- *Khả năng mở rộng:* Hệ thống này có khả năng mở rộng tốt, cho phép người dùng mở rộng cụm máy chủ để đáp ứng nhu cầu về dung lượng và hiệu suất.
- *Hỗ trợ truy vấn phức tạp:* Apache Doris hỗ trợ truy vấn phức tạp và linh hoạt, giúp người dùng thực hiện các phân tích phức tạp trên dữ liệu.
- *Tích hợp với các công cụ phân tích dữ liệu:* Apache Doris có thể tích hợp dễ dàng với các công cụ phân tích dữ liệu phổ biến như Tableau, Superset, và các công cụ BI khác.
- *Quản lý tài nguyên linh hoạt:* Hệ thống cho phép người quản trị linh hoạt quản lý tài nguyên, giúp tối ưu hóa việc sử dụng các nguồn lực hệ thống.

2.2 Nhược điểm

- *Hiệu suất ổn định:* Mặc dù Doris được thiết kế để đạt hiệu suất cao, nhưng có thể có những vấn đề liên quan đến hiệu suất ổn định trong một số trường hợp sử dụng cụ thể. Điều này có thể dẫn đến sự không ổn định hoặc giảm hiệu suất khi đối mặt với tải công việc cao.
- *Độ chậm trong việc cập nhật dữ liệu:* Trong môi trường yêu cầu cập nhật dữ liệu thường xuyên, Apache Doris có thể không hiệu quả bằng một số hệ thống khác. Việc xử lý các hoạt động cập nhật có thể làm giảm hiệu suất so với việc truy vấn dữ liệu.

- *Khả năng mở rộng*: Mặc dù Apache Doris có thể mở rộng theo chiều ngang để xử lý lớn dữ liệu, nhưng có thể tồn tại một số hạn chế về khả năng mở rộng trong môi trường cụ thể. Một số người dùng có thể gặp khó khăn khi triển khai và quản lý môi trường với quy mô lớn.

2.3 So sánh với các sản phẩm cùng loại

2.3.1 Apache Druid

Apache Druid là một hệ thống cơ sở dữ liệu phân tán mã nguồn mở được thiết kế để hỗ trợ phân tích dữ liệu thời gian thực và phức tạp. Nó có khả năng xử lý dữ liệu lớn, cung cấp khả năng truy vấn nhanh chóng và hỗ trợ cho các tình huống yêu cầu phân tích phức tạp.

- *Giống nhau*
 - Đều được thiết kế chủ yếu để hỗ trợ OLAP trên dữ liệu lớn, đặc biệt là truy vấn thời gian thực.
 - Cả hai hệ thống đều hỗ trợ kiến trúc phân tán và có khả năng mở rộng để xử lý dữ liệu lớn.
 - Hỗ trợ ngôn ngữ SQL cho việc truy vấn dữ liệu.
- *Khác nhau*
 - *Lưu trữ và định dạng dữ liệu*: Apache Doris hỗ trợ cả dữ liệu có cấu trúc và dữ liệu không cấu trúc, với việc lưu trữ dữ liệu dưới dạng hàng (columnar storage). Trong khi đó, Apache Druid chủ yếu sử dụng cấu trúc dữ liệu được tối ưu hóa cho việc truy vấn nhanh và lưu trữ dữ liệu dưới dạng cột.
 - *Mục tiêu sử dụng*: Trong khi Apache Druid tập trung chủ yếu vào việc phân tích dữ liệu thời gian thực và đặc biệt thích hợp

cho các ứng dụng như trực tuyến analytical processing (OLAP) trên dữ liệu log và sự kiện thời gian thực thì Apache Doris tập trung vào phân tích dữ liệu lớn với khả năng xử lý dữ liệu phức tạp và hỗ trợ cả OLAP và OLTP.

2.3.2 Apache Kylin

Apache Kylin là một hệ thống cơ sở dữ liệu mã nguồn mở được xây dựng để hỗ trợ phân tích OLAP (Online Analytical Processing) trên dữ liệu lớn. Được phát triển bởi Apache Software Foundation, Kylin chủ yếu được sử dụng để xử lý các truy vấn phức tạp trên dữ liệu đa chiều và cung cấp các tính năng OLAP mạnh mẽ.

- *Giống nhau*
 - Đề được thiết kế chủ yếu để hỗ trợ OLAP trên dữ liệu lớn, đặc biệt là truy vấn thời gian thực.
 - Có khả năng mở rộng và hoạt động trên nền tảng phân tán.
 - Hỗ trợ ngôn ngữ SQL cho việc truy vấn dữ liệu.
- *Khác nhau*
 - *Kiến trúc lưu trữ:* Apache Doris sử dụng kiến trúc lưu trữ dạng hàng (columnar storage). Trong khi đó, Apache Kylin sử dụng Apache Hbase hoặc Apache Parquet để lưu trữ dữ liệu.
 - *Hiệu suất và mở rộng:* Trong khi Kylin có thể phải đối mặt với vấn đề hiệu suất khi xử lý các Cube lớn và không mở rộng tốt trên mô hình dữ liệu thay đổi thường xuyên thì Apache Doris được thiết kế để có hiệu suất cao và khả năng mở rộng tốt trên cả đọc và ghi.

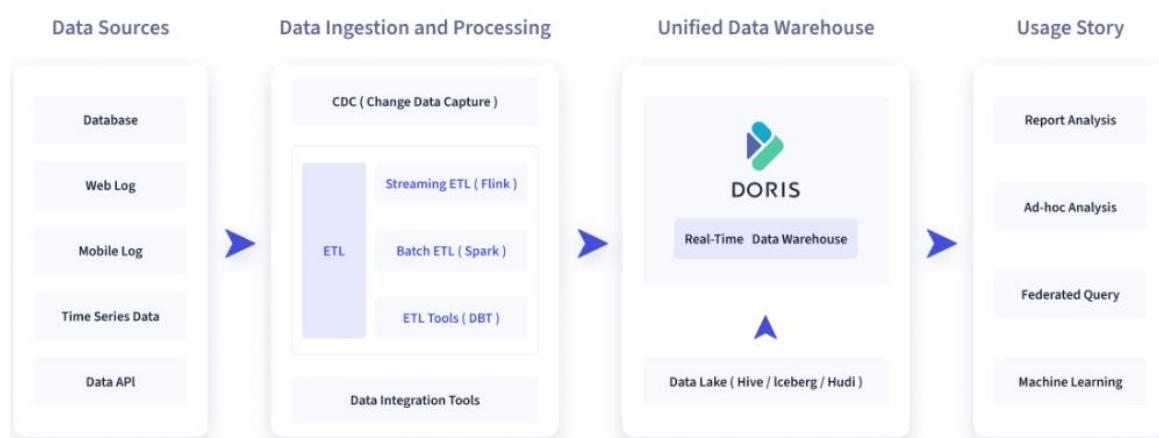
2.3.3 ClickHouse

ClickHouse là một cơ sở dữ liệu phân tích theo cột, mã nguồn mở được tạo bởi Yandex cho OLAP và các trường hợp sử dụng dữ liệu lớn. Với khả năng hỗ trợ xử lý truy vấn thời gian thực, ClickHouse phù hợp với các ứng dụng yêu cầu kết quả phân tích nhanh chóng.

- *Giống nhau*
 - Đề được thiết kế chủ yếu để hỗ trợ OLAP trên dữ liệu lớn, đặc biệt là truy vấn thời gian thực.
 - Có khả năng mở rộng và hoạt động trên nền tảng phân tán.
 - Cả hai đều có khả năng hỗ trợ xử lý dữ liệu thời gian thực, giúp đáp ứng nhanh chóng với thay đổi trong dữ liệu.
- *Khác nhau*
 - *Ngôn ngữ truy vấn:* ClickHouse sử dụng một ngôn ngữ truy vấn riêng, trong khi Apache Doris hỗ trợ SQL tiêu chuẩn, giúp ích cho việc tích hợp và sử dụng.
 - *Định dạng dữ liệu:* ClickHouse thường được ưu tiên cho các công việc xử lý dữ liệu có cấu trúc đơn giản và dữ liệu không cấu trúc. Trong khi đó, Apache Doris có thể xử lý các cấu trúc dữ liệu phức tạp hơn.

CHƯƠNG 3: CÁC TRƯỜNG HỢP CỤ THỂ ỨNG DỤNG APACHE DORIS

Như trong hình bên dưới, sau khi tích hợp và xử lý dữ liệu khác nhau, các nguồn dữ liệu thường được lưu trữ trong kho dữ liệu thời gian thực Apache Doris và hồ dữ liệu ngoại tuyến hoặc kho dữ liệu (trong Apache Hive, Apache Iceberg hoặc Apache Hudi).



3.1 Phân tích báo cáo

Apache Doris có thể được sử dụng để tạo các trang tổng quan thời gian thực, báo cáo cho các nhà phân tích và quản lý nội bộ, cũng như các báo cáo hướng người dùng hoặc hướng khách hàng đồng thời cao. Ví dụ: JD.com sử dụng Doris để tạo báo cáo quảng cáo, với hàng chục nghìn truy vấn đồng thời và độ trễ truy vấn dưới giây.

3.2 Truy vấn Ad-hoc

Apache Doris có thể được sử dụng để thực hiện các truy vấn tự phục vụ theo định hướng của nhà phân tích với các mẫu truy vấn bất thường và yêu cầu thông lượng cao. Ví dụ: XiaoMi đã sử dụng Doris để phân tích dữ liệu hành vi của người dùng, với độ trễ truy vấn trung bình là 10 giây và độ trễ truy vấn phân vị thứ 95 là 30 giây trở xuống.

3.3 Xây dựng kho dữ liệu đồng nhất

Apache Doris có thể được sử dụng để xây dựng một nền tảng đáp ứng nhu cầu xây dựng kho dữ liệu thống nhất và đơn giản hóa ngăn xếp phần mềm dữ liệu phức tạp. Ví dụ: HaiDiLao đã sử dụng Doris để thay thế kiến trúc cũ bao gồm Apache Spark, Apache Hive, Apache Kudu, Apache HBase và Apache Phoenix.

3.4 Tăng tốc độ truy vấn Data Lake

Apache Doris có thể được sử dụng để truy vấn dữ liệu nằm trong Apache Hive, Apache Iceberg và Apache Hudi bằng cách sử dụng các bảng bên ngoài. Điều này giúp cải thiện đáng kể hiệu suất truy vấn mà không cần sao chép dữ liệu.

CHƯƠNG 4: HƯỚNG DẪN CÀI ĐẶT VÀ CẤU HÌNH

4.1 Cài đặt Apache Doris

Môi trường mà nhóm đã cài đặt:

- Ubuntu 20.04 trở lên.
- Java Development Kit (Phiên bản JDK 11.0.20).

Để kiểm tra phiên bản Java bạn đã cài đặt, hãy chạy lệnh sau.

```
Java -version
```

Tiếp theo tải xuống [file FE](#) và [file BE](#) (Phiên bản 1.1.4) và đồng thời giải nén nó.

```
Tar xzf apache-doris-x.x.x.tar.gz
```

4.2 Cấu hình Apache Doris

4.2.1 Cấu hình Frontend (FE)

Đi đến thư mục *apache-doris-x.x.x/fe*

```
cd apache-doris-x.x.x/fe
```

Sửa file cấu hình FE *conf/fe.conf*, ở đây chủ yếu sửa 2 parameters là *priority_networks* và *meta_dir*, nếu cần cấu hình tối ưu hơn có thể tham khảo phần [cấu hình thông số FE](#) để được hướng dẫn chỉnh sửa.

- Thêm parameter *priority_networks*

```
priority_networks=172.23.16.0/24;"IP máy"
```

* **Lưu ý:** Thông số này chúng ta phải cấu hình trong quá trình cài đặt, đặc biệt khi một máy có nhiều địa chỉ IP thì chúng ta phải chỉ định một địa chỉ IP duy nhất cho FE.

- Thêm thư mục chứa siêu dữ liệu

```
meta_dir=/path/your/doris-meta
```

* **Lưu ý:** Ở đây có thể để không cấu hình, mặc định là thư mục doris-meta có sẵn trong thư mục cài đặt Doris FE. Nếu muốn cấu hình thư mục siêu dữ liệu riêng, thì chúng ta cần tạo thư mục chỉ định trước.

- Start FE:

Thực hiện lệnh sau trong thư mục cài đặt FE để hoàn tất quá trình khởi động FE.

```
./bin/start_fe.sh --daemon
```

Xem trạng thái hoạt động của FE, có thể kiểm tra xem Doris đã khởi động thành công chưa bằng lệnh sau:

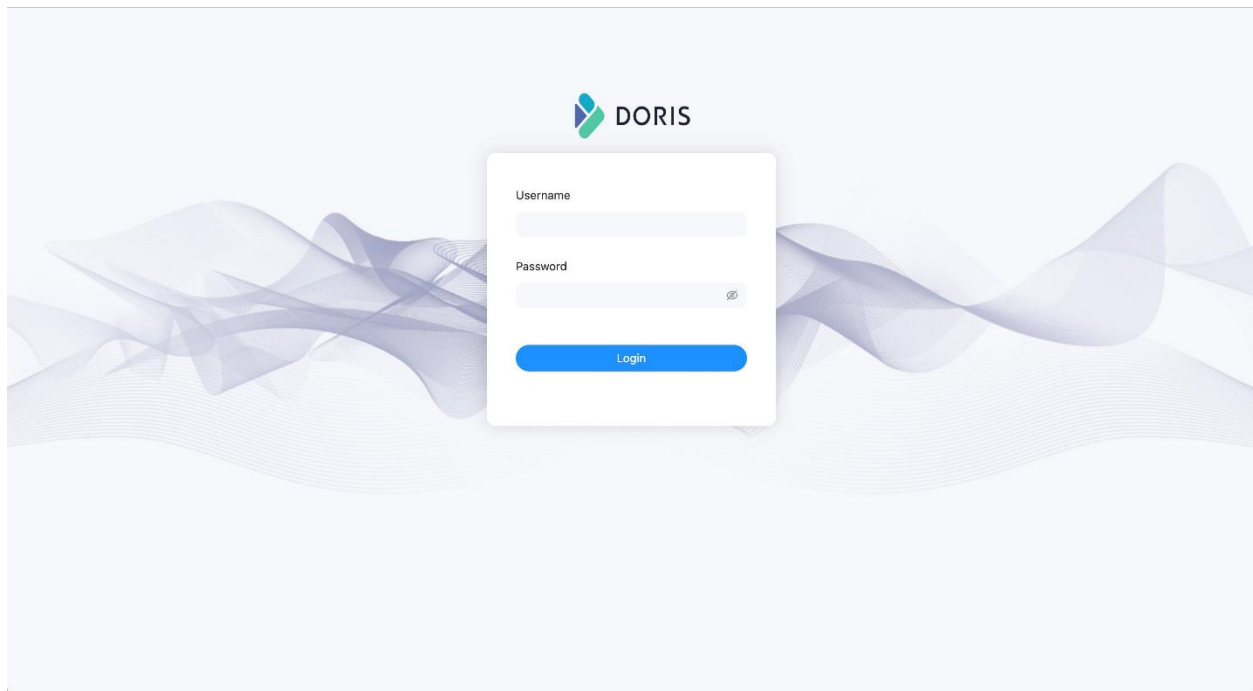
```
curl http://127.0.0.1:8030/api/bootstrap
```

Ở đây IP và port là IP và http_port của FE (mặc định 8030), nếu đang thực hiện trong node FE thì chỉ cần chạy trực tiếp câu lệnh trên. Nếu kết quả trả là "*msg*": "*success*" thì khởi động thành công.

Chúng ta cũng có thể kiểm tra điều này thông qua giao diện người dùng web do Doris FE cung cấp bằng cách truy cập vào địa chỉ sau.

```
http:// fe_ip:8030
```

Chúng ta có thể thấy màn hình sau, cho biết FE đã khởi động thành công.



* **Lưu ý:** Ở đây, sử dụng người dùng mặc định tích hợp sẵn của Doris, root, để đăng nhập bằng mật khẩu trống. Đây là giao diện quản trị dành riêng cho những người dùng có quyền quản trị.

- Connect FE

Tiếp theo sẽ kết nối với Doris FE thông qua MySQL client. Đầu tiên cần tải xuống [MySQL Client](#) và giải nén nó. Sau đó, bạn có thể tìm thấy công cụ dòng lệnh mysql trong thư mục bin/ và thực hiện lệnh sau để kết nối với Doris.

```
mysql -uroot -P9030 -h127.0.0.1
```

* **Lưu ý:** Người dùng root được sử dụng ở đây là người dùng mặc định được tích hợp trong doris và cũng là người dùng quản trị viên cấp cao.

- -P: Đây là cổng truy vấn của chúng ta để kết nối với Doris, cổng mặc định là 9030, tương ứng với query_port trong fe.conf

- -h: Đây là địa chỉ IP của FE mà chúng ta đang kết nối, nếu máy khách của bạn và FE được cài đặt trên cùng một nút, bạn có thể sử dụng 127.0.0.1, đây cũng là do Doris cung cấp nếu bạn quên mật khẩu root, bạn có thể kết nối trực tiếp vào đăng nhập mà không cần mật khẩu theo cách này và đặt lại mật khẩu gốc.

Thực hiện lệnh sau để xem trạng thái chạy FE:

```
show frontends\G;
```

Kết quả:

```
mysql> show frontends\G;

***** 1. Row
*****

      Name: 172.21.32.5_9010_1660549353220
      IP: 172.21.32.5
EditLogPort: 9010
  HttpPort: 8030
  QueryPort: 9030
    RpcPort: 9020
      Role: FOLLOWER
  IsMaster: true
ClusterId: 1685821635
      Join: true
    Alive: true
```

```
ReplayedJournalId: 49292
      LastHeartbeat: 2022-08-17 13:00:45
            IsHelper: true
                  ErrMsg:
                        Version: 1.1.2-rc03-ca55ac2
CurrentConnected: Yes
1 row in set (0.03 sec)
```

Nếu các cột IsMaster, Join và Alive có giá trị **true** thì node đó đang hoạt động bình thường.

- Stop FE

Việc dừng Doris FE có thể được thực hiện bằng lệnh sau:

```
./bin/stop_fe.sh
```

4.2.2 Cấu hình Backend (BE)

Chuyển đến thư mục **apache-doris-x.x.x/be**

```
cd apache-doris-x.x.x/be
```

Sửa file cấu hình BE conf/be.conf, ở đây chủ yếu sửa 2 parameters **priority_networks** và **storage_root**, nếu bạn cần cấu hình tối ưu hơn có thể tham khảo hướng dẫn [cấu hình thông số BE](#) để điều chỉnh.

- Thêm parameter **priority_networks**

```
priority_networks=172.23.16.0/24;"IP máy"
```

* **Lưu ý:** Thông số này chúng ta phải cấu hình trong quá trình cài đặt, đặc biệt khi một máy có nhiều địa chỉ IP thì chúng ta phải gán một địa chỉ IP duy nhất cho BE.

- Thêm đường dẫn JAVA_HOME:

```
JAVA_HOME="/usr/lib/jvm/jdk-11.0.21"
```

- Configure thư mục lưu trữ dữ liệu BE

```
storage_root_path=/path/your/data_dir
```

* **Lưu ý:** Thư mục mặc định nằm trong thư mục lưu trữ của thư mục cài đặt BE. Thư mục lưu trữ cho cấu hình BE phải được tạo trước.

- Start BE

Thực hiện lệnh sau trong thư mục cài đặt BE để hoàn thành khởi động BE

```
./bin/start_be.sh --daemon
```

Thêm một nút BE vào một cụm bằng cách kết nối với FE thông qua MySQL client và thực thi SQL sau để thêm BE vào cụm.

```
ALTER SYSTEM ADD BACKEND  
"be_host_ip:heartbeat_service_port";
```

- *be_host_ip*: Đây là địa chỉ IP BE của bạn, khớp với *priority_networks* trong *be.conf*.
- *heartbeat_service_port*: Đây là cổng tải lên Heartbeat của BE của bạn, khớp với *Heartbeat_service_port* trong *be.conf*, mặc định là 9050.

- Xem trạng thái hoạt động BE

Có thể kiểm tra trạng thái chạy của BE bằng cách thực hiện lệnh sau tại dòng lệnh MySQL:

```
SHOW BACKENDS\G;
```

Kết quả:

```
mysql> SHOW BACKENDS\G;

***** 1. row
*****

      BackendId: 10003
      Cluster: default_cluster
      IP: 172.21.32.5
      HeartbeatPort: 9050
      BePort: 9060
      HttpPort: 8040
      BrpcPort: 8060
      LastStartTime: 2022-08-16 15:31:37
      LastHeartbeat: 2022-08-17 13:33:17
      Alive: true
      SystemDecommissioned: false
      ClusterDecommissioned: false
      TabletNum: 170
      DataUsedCapacity: 985.787 KB
      AvailCapacity: 782.729 GB
```

```
TotalCapacity: 984.180 GB
UsedPct: 20.47 %
MaxDiskUsedPct: 20.47 %
Tag: {"location" : "default"}
ErrMsg:
Version: 1.1.2-rc03-ca55ac2
Status:
{"lastSuccessReportTabletsTime":"2022-08-17
13:33:05","lastStreamLoadTime":-
1,"isQueryDisabled":false,"isLoadDisabled":false}
1 row in set (0.01 sec)
```

Nếu cột Alive có giá trị *true* thì node đang hoạt động bình thường.

- Stop BE

Việc dừng Doris BE có thể được thực hiện bằng lệnh sau:

```
./bin/stop_be.sh
```

4.3 Minh họa truy vấn dữ liệu

4.3.1 Mô tả dữ liệu

Tạo database Demo:

```
Create database demo;

Tạo bảng example_tbl:

use demo;
```

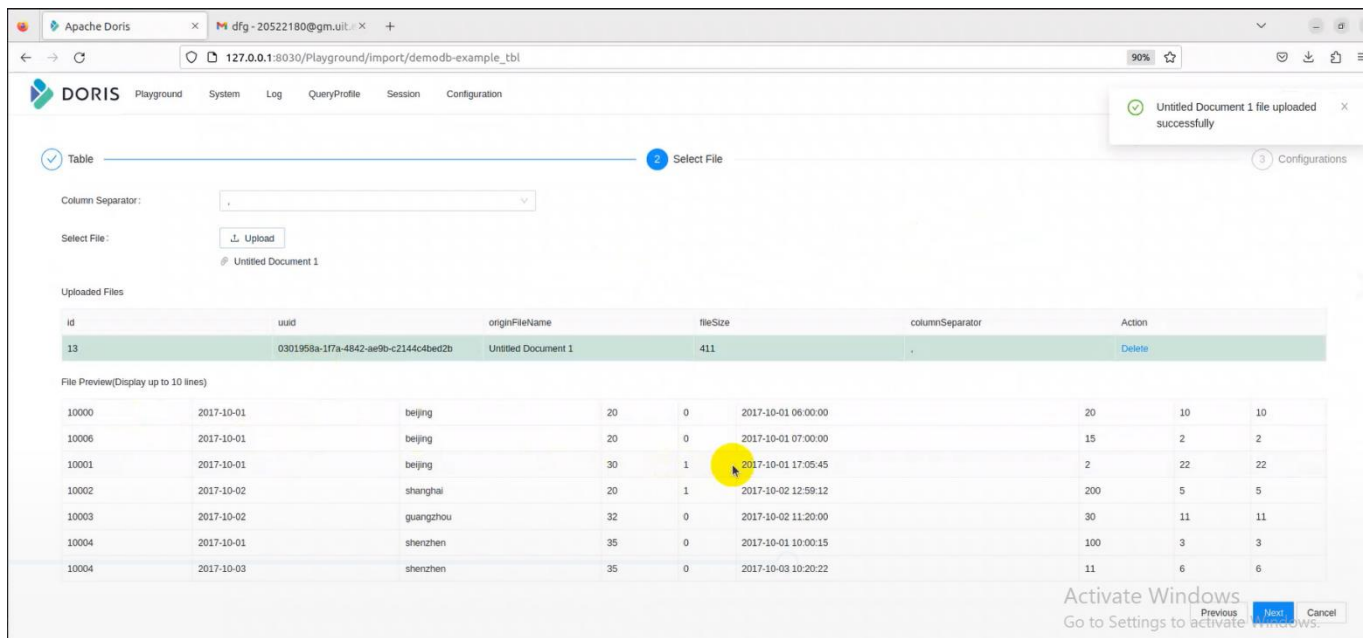
```
CREATE TABLE IF NOT EXISTS demo.example_tbl
(
    `user_id` LARGEINT NOT NULL COMMENT "user id",
    `date` DATE NOT NULL COMMENT "",
    `city` VARCHAR(20) COMMENT "",
    `age` SMALLINT COMMENT "",
    `sex` TINYINT COMMENT "",
    `last_visit_date` DATETIME REPLACE DEFAULT "1970-01-01 00:00:00" COMMENT "",
    `cost` BIGINT SUM DEFAULT "0" COMMENT "",
    `max_dwell_time` INT MAX DEFAULT "0" COMMENT "",
    `min_dwell_time` INT MIN DEFAULT "99999" COMMENT ""
)
AGGREGATE KEY(`user_id`, `date`, `city`, `age`, `sex`)
DISTRIBUTED BY HASH(`user_id`) BUCKETS 1
PROPERTIES (
    "replication_allocation" = "tag.location.default: 1"
);
```

Dữ liệu được lưu trữ trong file text có dạng như sau:

```
10000,2017-10-01,beijing,20,0,2017-10-01
06:00:00,20,10,10
```

```
10006,2017-10-01,beijing,20,0,2017-10-01
07:00:00,15,2,2
10001,2017-10-01,beijing,30,1,2017-10-01
17:05:45,2,22,22
10002,2017-10-02,shanghai,20,1,2017-10-02
12:59:12,200,5,5
10003,2017-10-02,guangzhou,32,0,2017-10-02
11:20:00,30,11,11
10004,2017-10-01,shenzhen,35,0,2017-10-01
10:00:15,100,3,3
10004,2017-10-03,shenzhen,35,0,2017-10-03
10:20:22,11,6,6
```

Thực hiện import dữ liệu trong giao diện Doris: Trong giao diện Doris Playground, chọn table trong database cần import dữ liệu, chọn Data import:



Table

Column Separator: ,

Select File: [Upload](#)

Uploaded Files

id	uuid	originFileName	fileSize	columnSeparator	Action
13	0301958a-1f7a-4842-ae9b-c2144c4bed2b	Untitled Document 1	411	,	Delete

File Preview (Display up to 10 lines)

id	uuid	originFileName	fileSize	columnSeparator	Action
10000	2017-10-01	beijing	20	0	2017-10-01 06:00:00
10006	2017-10-01	beijing	20	0	2017-10-01 07:00:00
10001	2017-10-01	beijing	30	1	2017-10-01 17:05:45
10002	2017-10-02	shanghai	20	1	2017-10-02 12:59:12
10003	2017-10-02	guangzhou	32	0	2017-10-02 11:20:00
10004	2017-10-01	shenzhen	35	0	2017-10-01 10:00:15
10004	2017-10-03	shenzhen	35	0	2017-10-03 10:20:22

4.3.2 Truy vấn dữ liệu

Sử dụng cú pháp SQL để thực hiện truy vấn dữ liệu:

- Truy vấn dữ liệu của bảng example_tbl:

```
Select * from example_tbl;
```

Results

Run successfully

select * from example_tbl;

Execution Time: 60 ms

user_id	date	city	age	sex	last_visit_date	cost	max_dwell_time	min_dwell_time
10000	2017-10-01	beijing	20	0	2017-10-01 06:00:00	20	10	10
10001	2017-10-01	beijing	30	1	2017-10-01 17:05:45	2	22	22
10002	2017-10-02	shanghai	20	1	2017-10-02 12:59:12	200	5	5
10003	2017-10-02	guangzhou	32	0	2017-10-02 11:20:00	30	11	11
10004	2017-10-01	shenzhen	35	0	2017-10-01 10:00:15	100	3	3
10004	2017-10-03	shenzhen	35	0	2017-10-03 10:20:22	11	6	6
10006	2017-10-01	beijing	20	0	2017-10-01 07:00:00	15	2	2

```
Select * from example_tbl where sex=1;
```

Results

Run successfully

select * from example_tbl where sex=1;

Execution Time: 125 ms

user_id	date	city	age	sex	last_visit_date	cost	max_dwell_time	min_dwell_time
10001	2017-10-01	beijing	30	1	2017-10-01 17:05:45	2	22	22
10002	2017-10-02	shanghai	20	1	2017-10-02 12:59:12	200	5	5

```
Select city, sum(cost) as total_cost from example_tbl  
group by city;
```

Results

Run successfully

```
select city, sum(cost) as total_cost from example_tbl group by city;
```

Execution Time: 116 ms

city	total_cost
beijing	37
shanghai	200
guangzhou	30
shenzhen	111

NGUỒN THAM KHẢO

- [1] "Doris," [Online]. Available: <https://doris.apache.org/docs/1.2/get-starting/>.
- [2] "Single instruction, multiple data," [Online]. Available:
https://en.wikipedia.org/wiki/Single_instruction,_multiple_data.
- [3] O. Peckham, "Apache Doris Analytical Database Graduates from Apache Incubator," [Online]. Available:
<https://www.datanami.com/2022/06/20/apache-doris-analytical-database-graduates-from-apache-incubator/>. [Accessed 20 June 2022].