

25 YEARS ANNIVERSARY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY



HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Chương 4

NoSQL - phần 3

Công cụ xử lý truy vấn

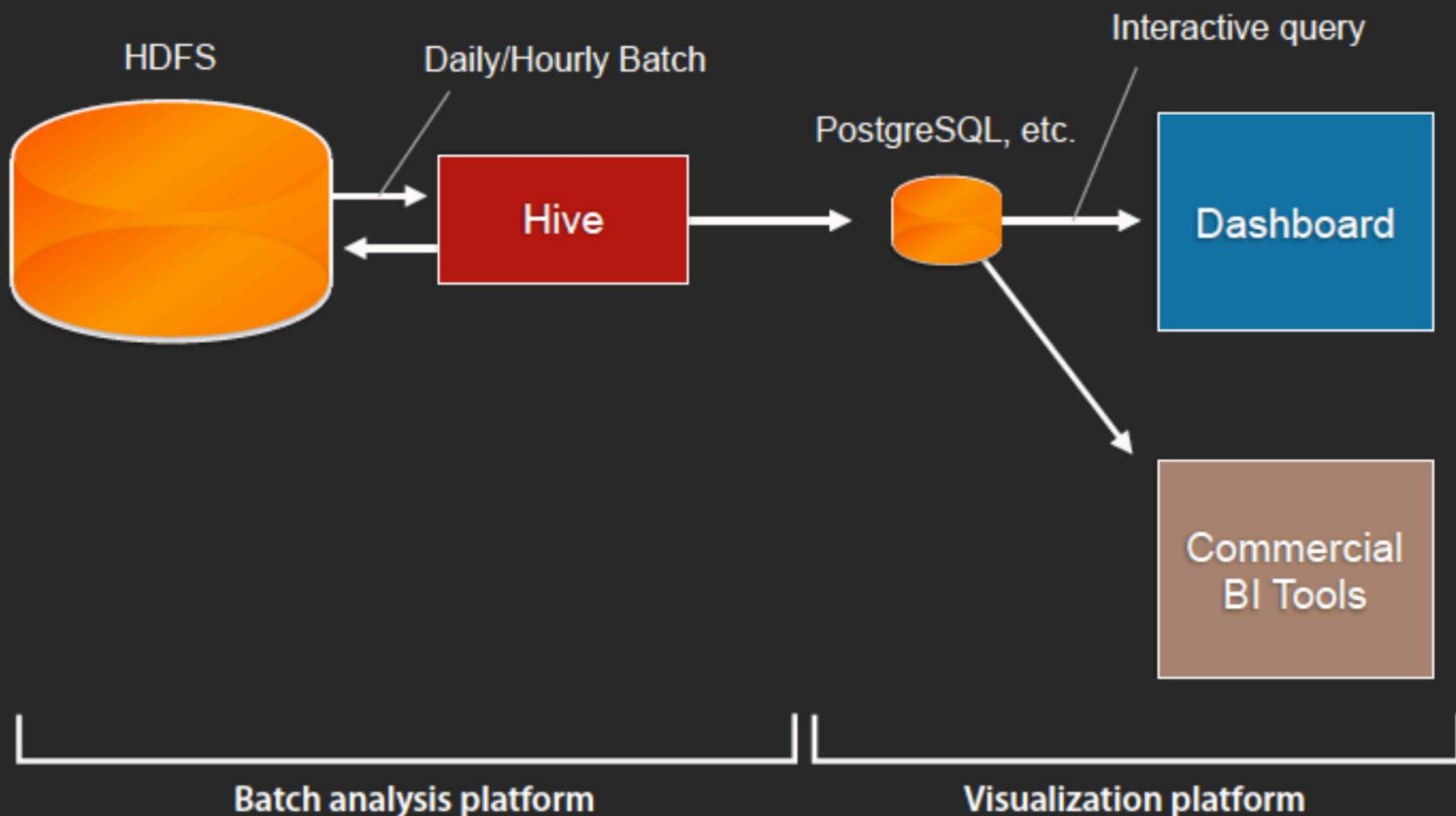
Lịch sử

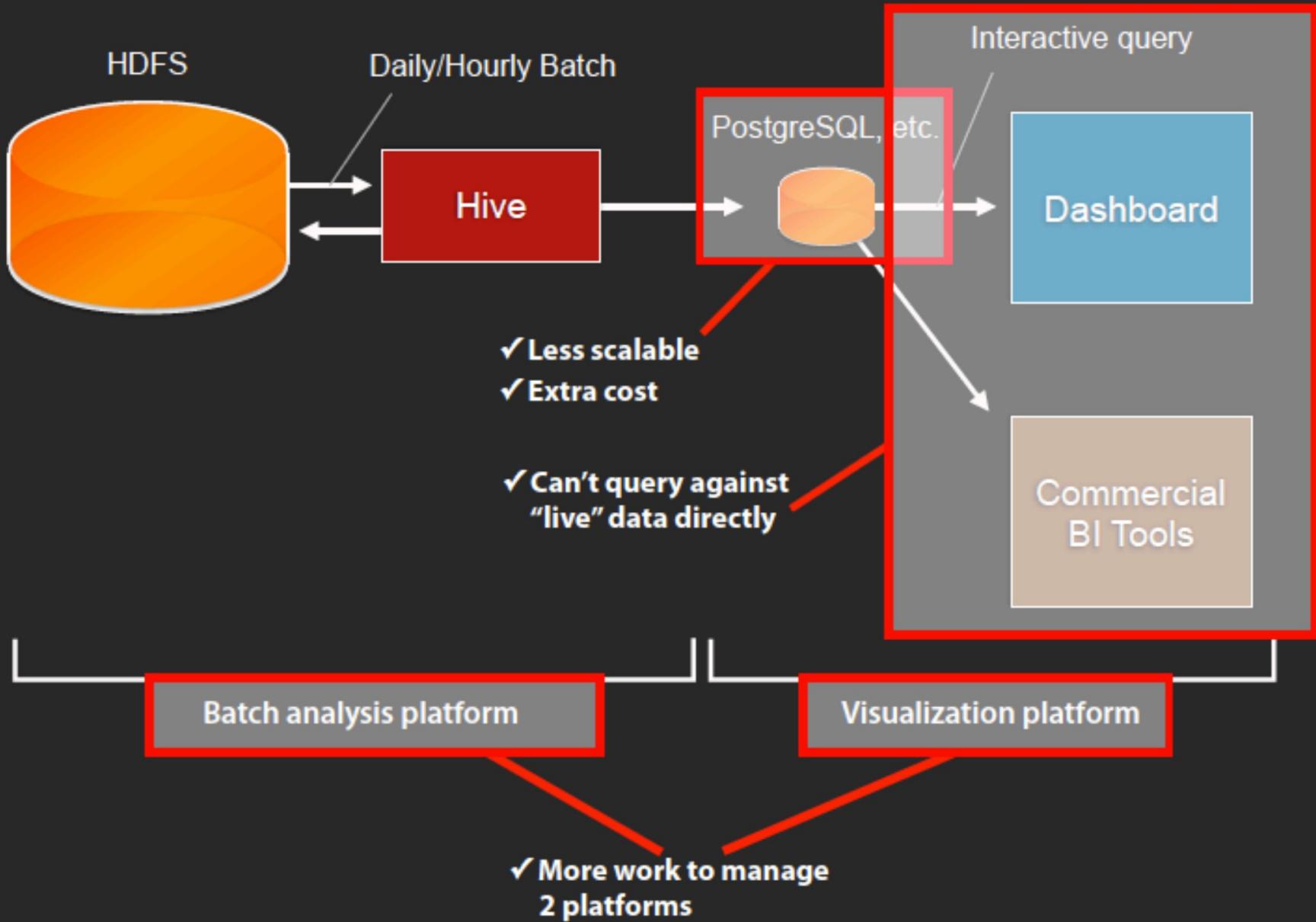
- Mùa thu năm 2012: Dự án bắt đầu tại Facebook • Được thiết kế cho truy vấn tương tác • với tốc độ của kho dữ liệu thương mại • và khả năng mở rộng theo quy mô của Facebook • Mùa đông năm 2013: Nguồn mở
- Hơn 30 người đóng góp trong 6 tháng • bao gồm những người bên ngoài Facebook
- Năm 2019: Hơn 300 người đóng góp

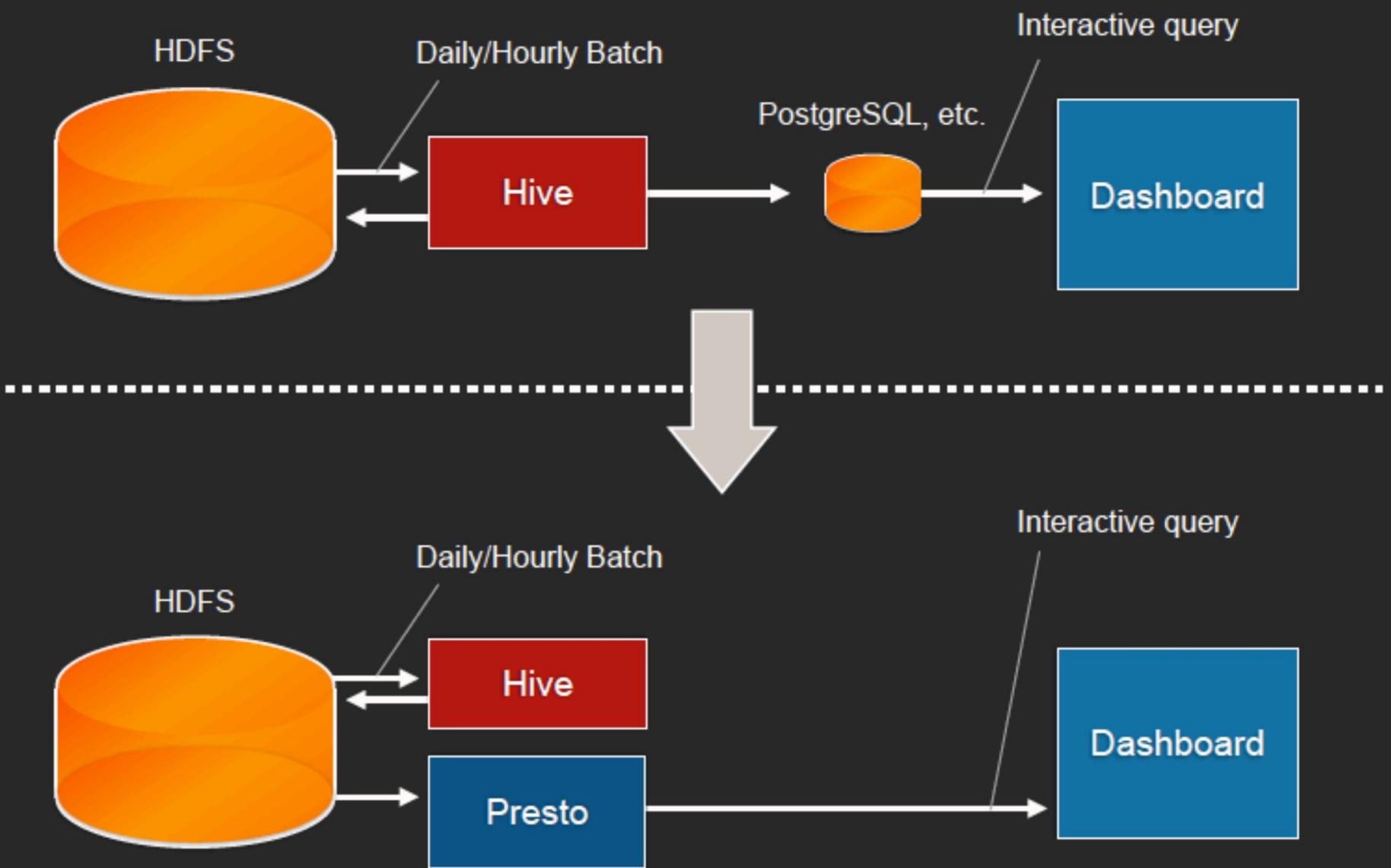
Động lực

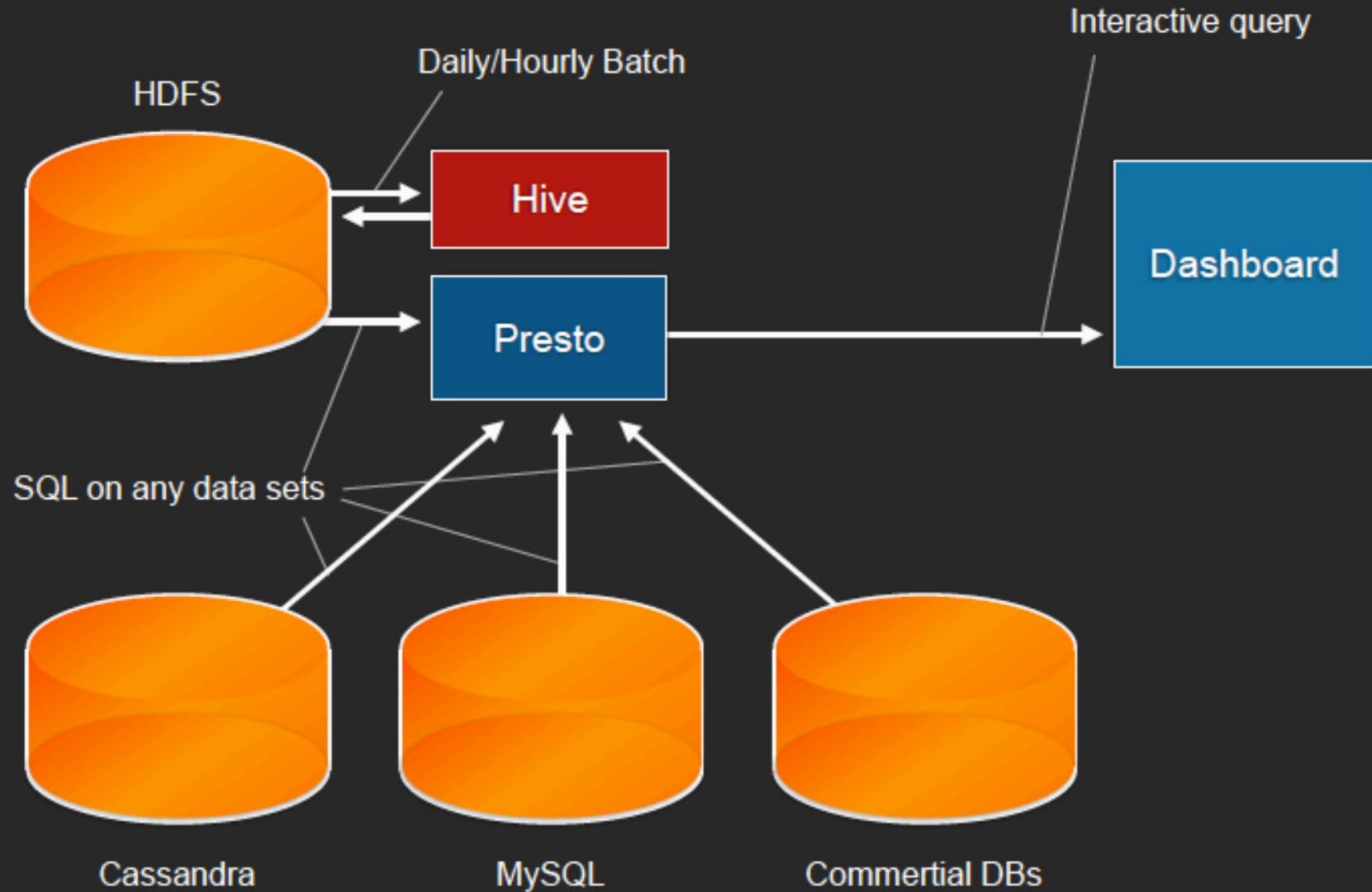
- Chúng tôi không thể trực quan hóa dữ liệu trong HDFS bằng bảng điều khiển hoặc công cụ BI
 - vì Hive quá chậm (không tương tác) • hoặc kết nối ODBC không khả dụng/không ổn định • Chúng tôi cần lưu trữ kết quả hàng loạt hàng ngày vào một DB tương tác để phản hồi nhanh (PostgreSQL, Redshift, v.v.) • DB tương tác tồn kém hơn nhiều nhưng khả năng mở rộng lại kém hơn nhiều • Một số dữ liệu không được lưu trữ trong HDFS
 - Chúng ta cần sao chép dữ liệu vào HDFS để phân tích

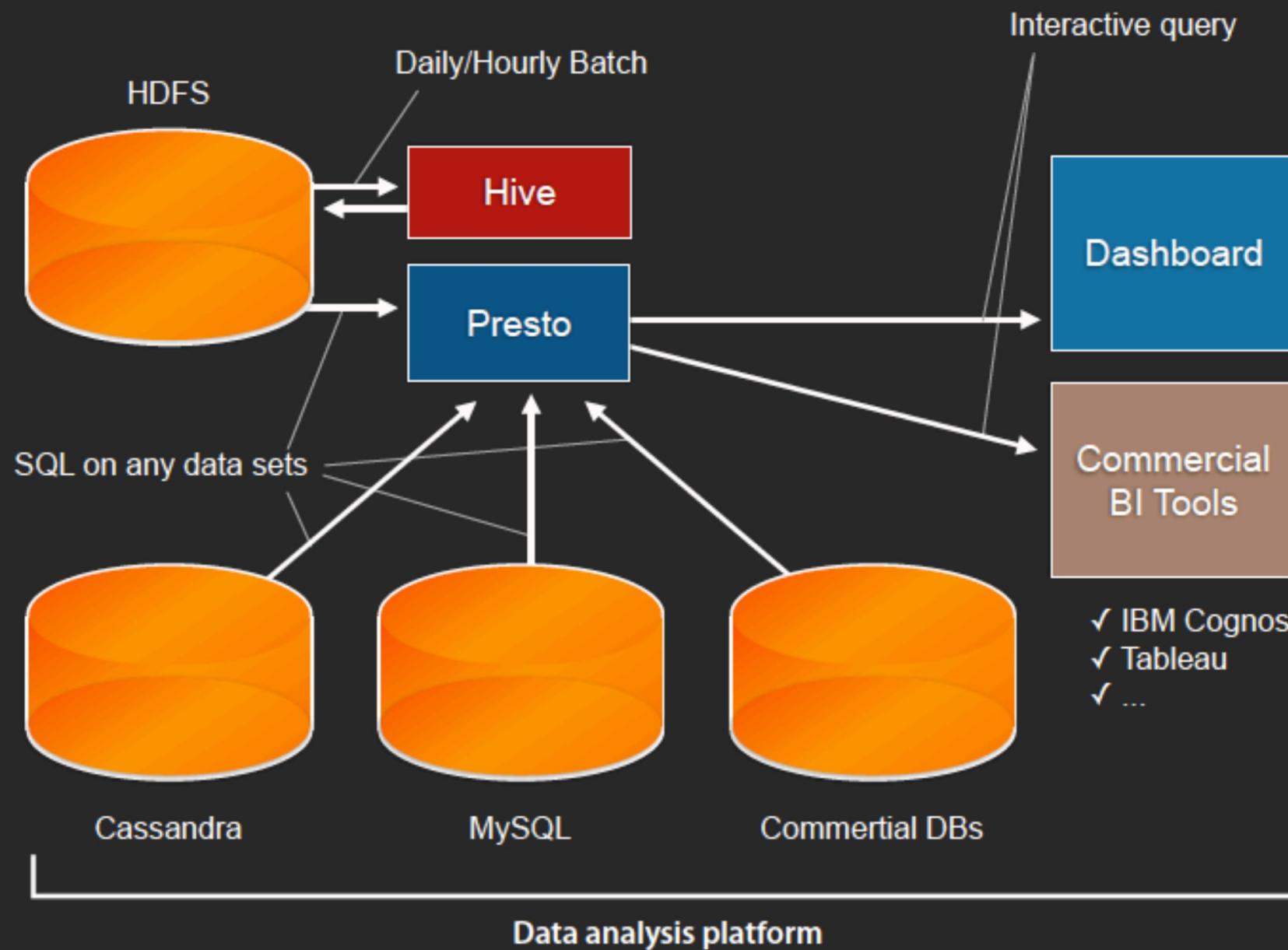
khả năng trích xuất thông tin chi tiết nhanh chóng và dễ dàng từ lượng dữ liệu lớn











Presto có thể làm gì?

- Công cụ truy vấn SQL phân tán nguồn mở đã chạy trong sản xuất tại Facebook kể từ năm 2013
 - Giao diện ANSI SQL
- Truy vấn tương tác (tính bằng mili giây đến phút)
 - MapReduce và Hive vẫn cần thiết cho ETL
- Truy vấn bằng các công cụ BI thương mại hoặc bảng thông tin
 - Kết nối ODBC/JDBC đáng tin cậy
- Truy vấn trên nhiều nguồn dữ liệu như Hive, HBase, Cassandra hoặc thậm chí là DB thương mại
 - Cơ chế plugin
- Tích hợp phân tích hàng loạt + trực quan hóa vào một nền tảng phân tích dữ liệu duy nhất

Triển khai Presto

- Facebook (2013) •

Nhiều khu vực địa lý • Mở rộng

lên 1.000 nút • Được hơn

1.000 nhân viên sử dụng tích cực, chạy hơn 30.000 truy vấn
mỗi ngày

- Xử lý 1PB/ngày



NETFLIX



facebook



LinkedIn



TERADATA®



Amazon Athena



FreeWheel



Bloomberg

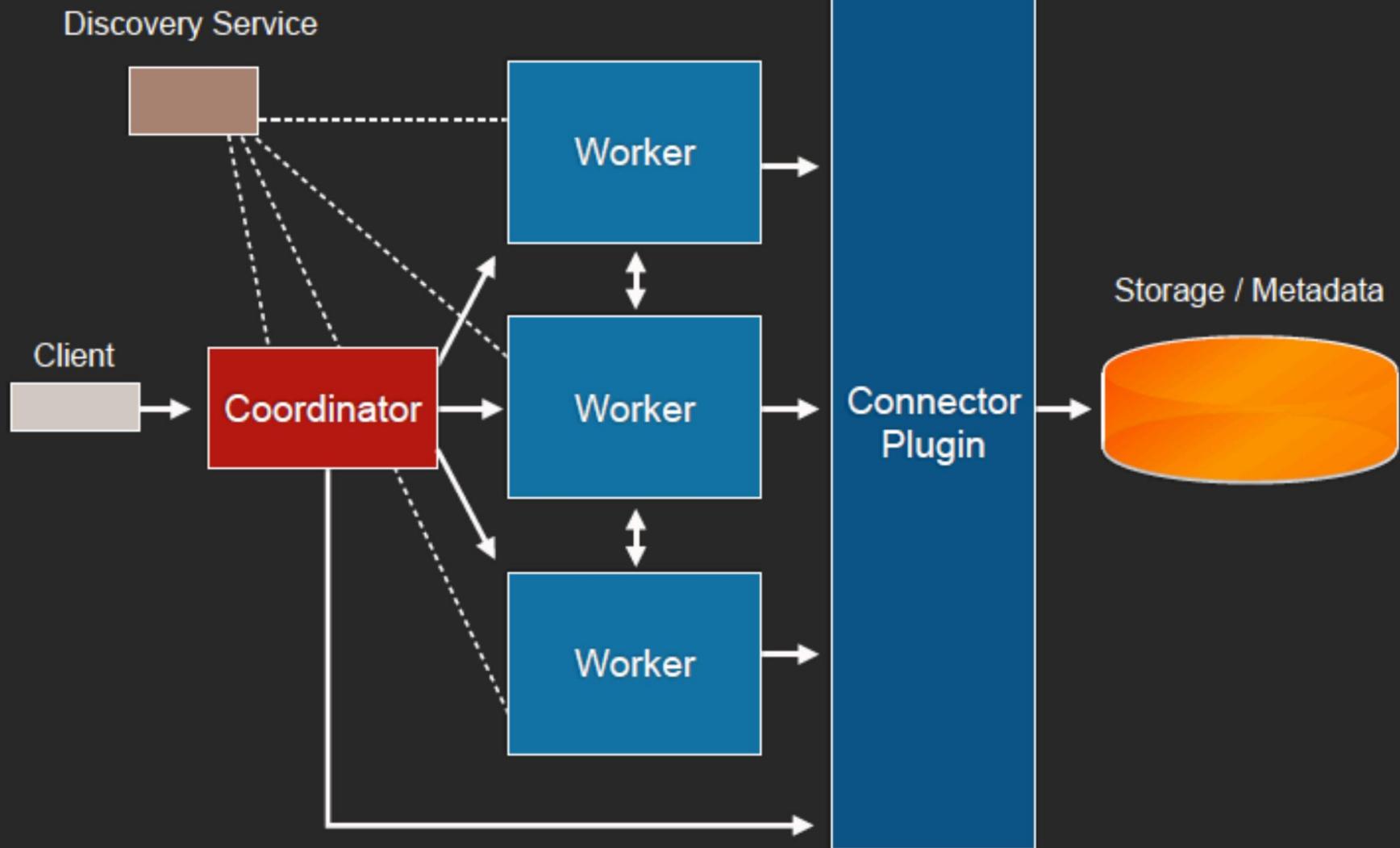


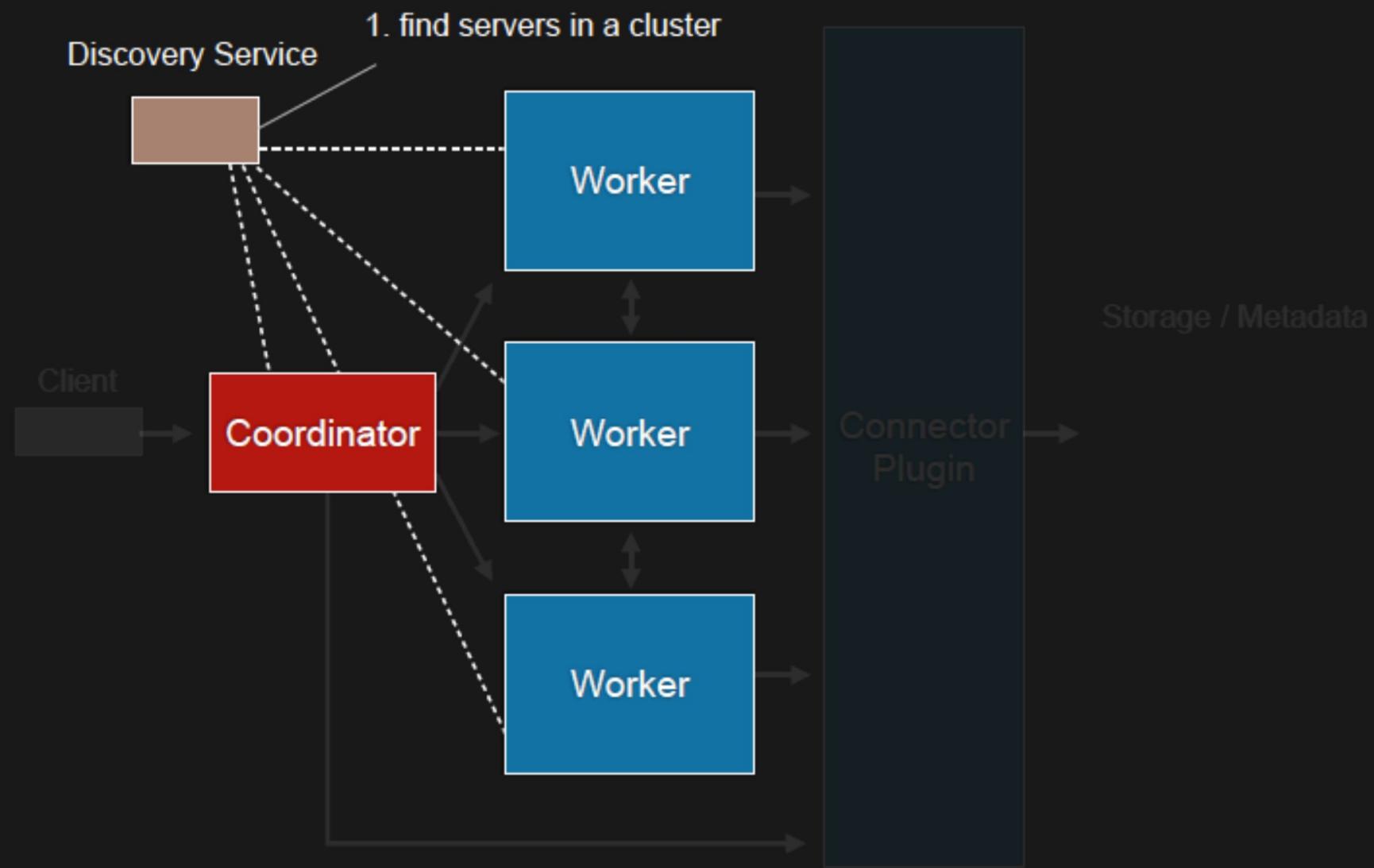
Pinterest

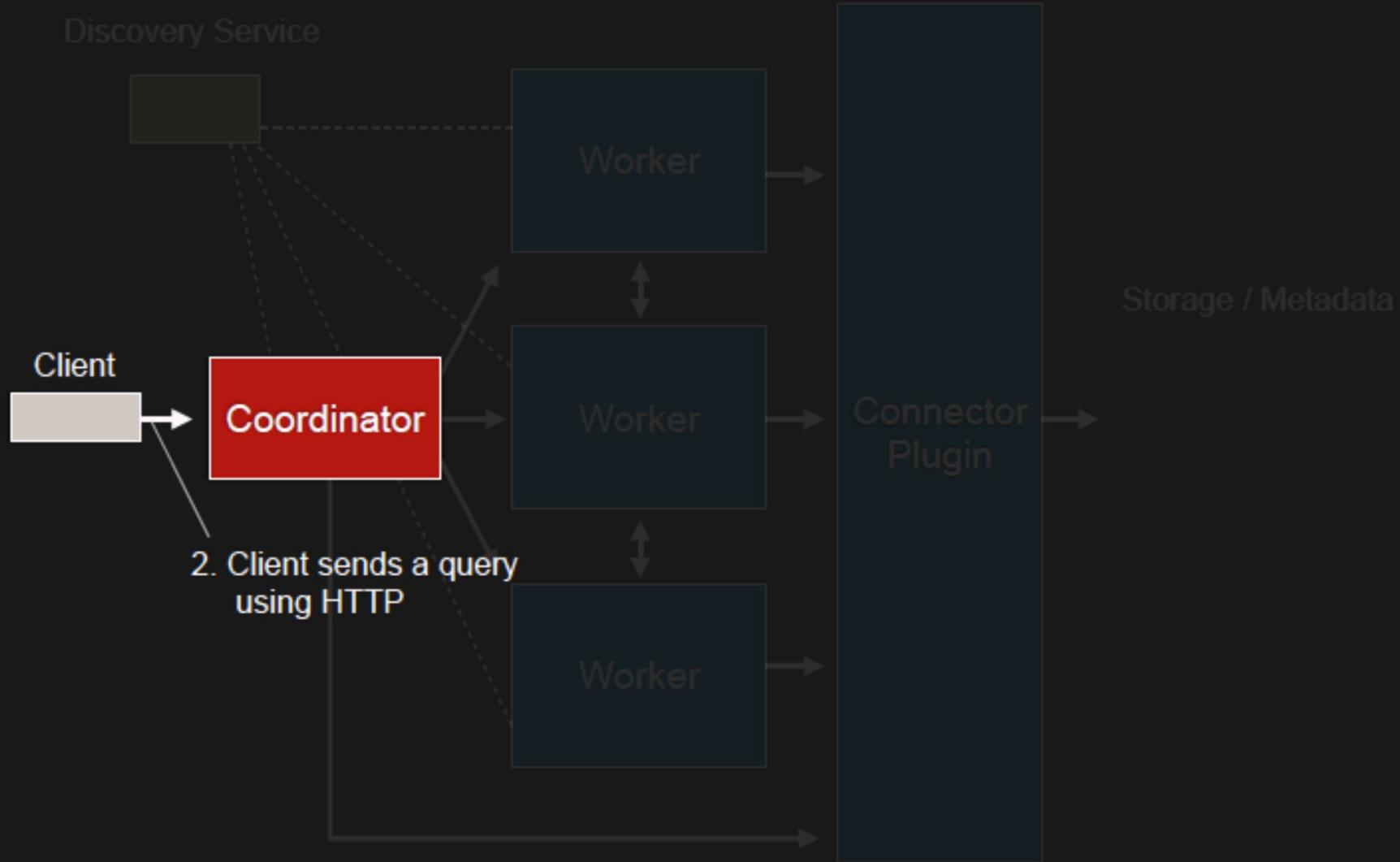


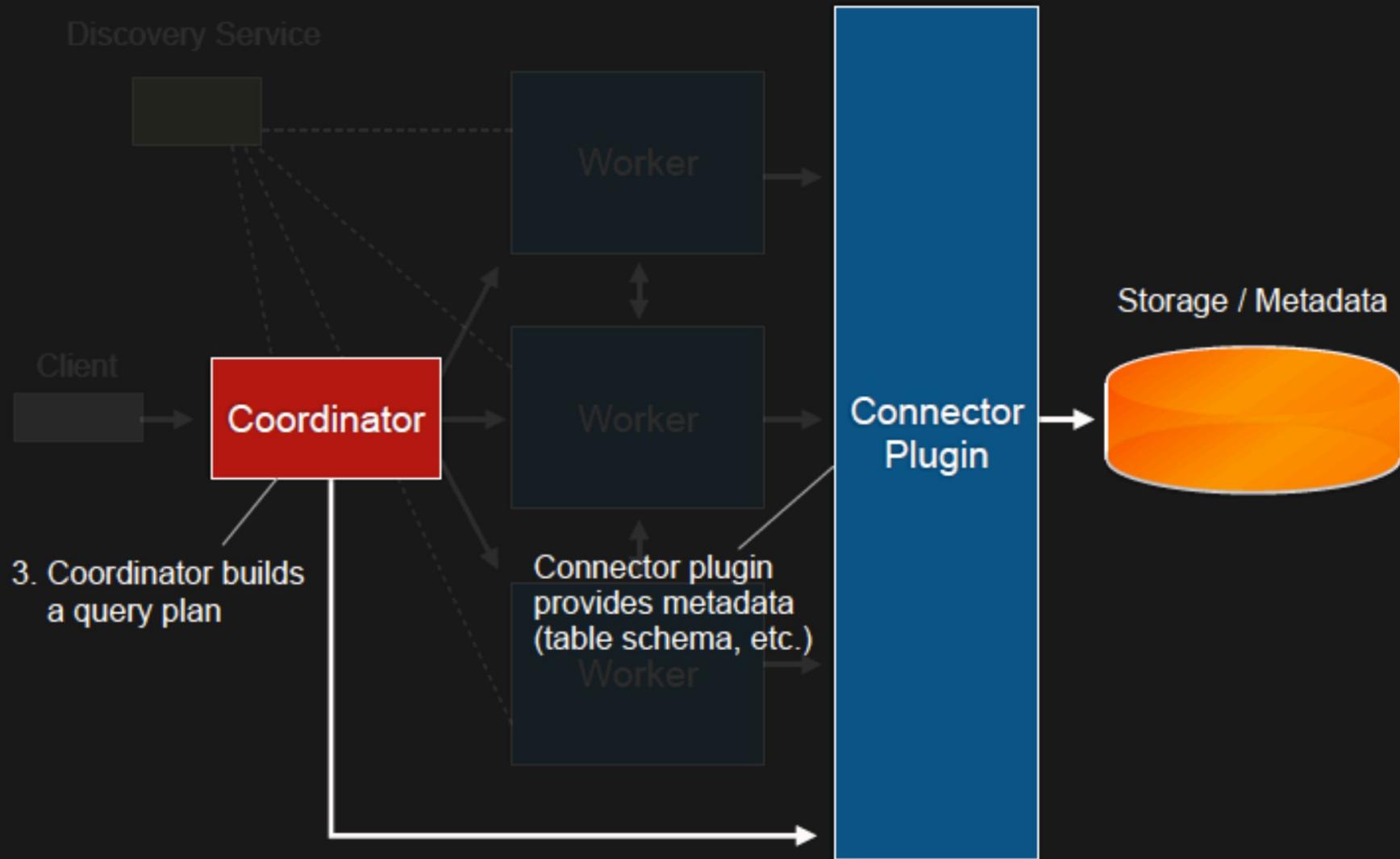
NETFLIX

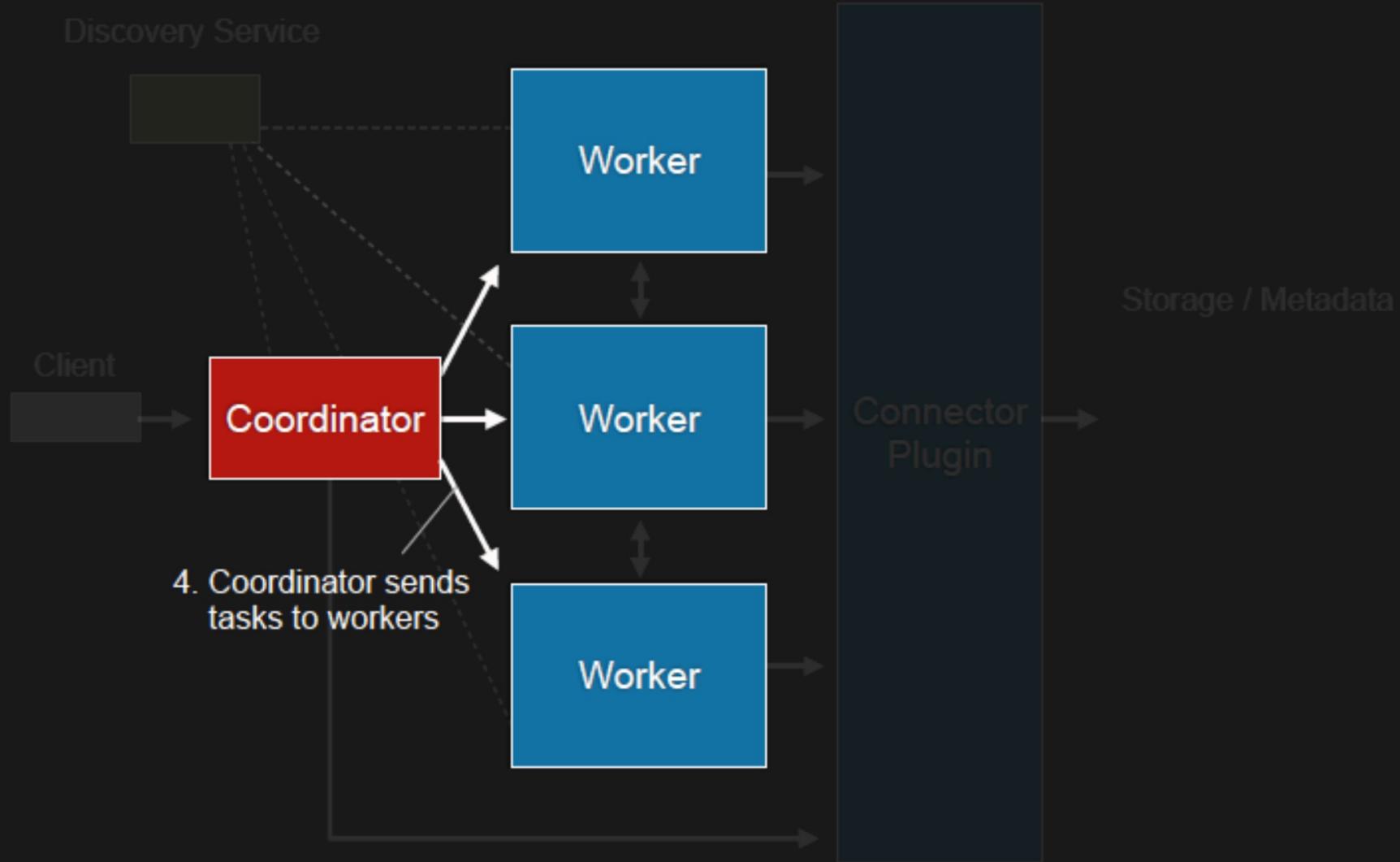
Kiến trúc Presto

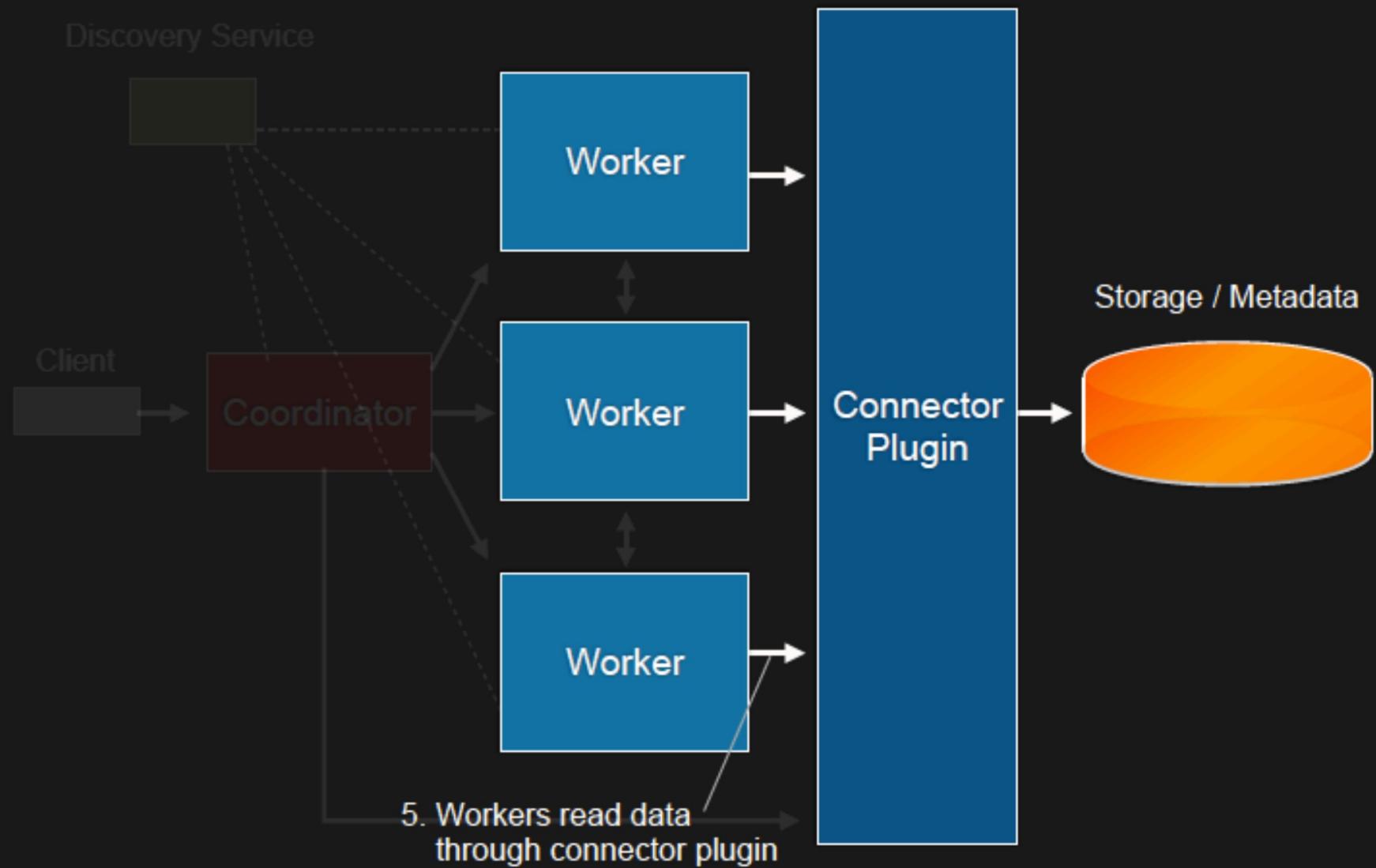


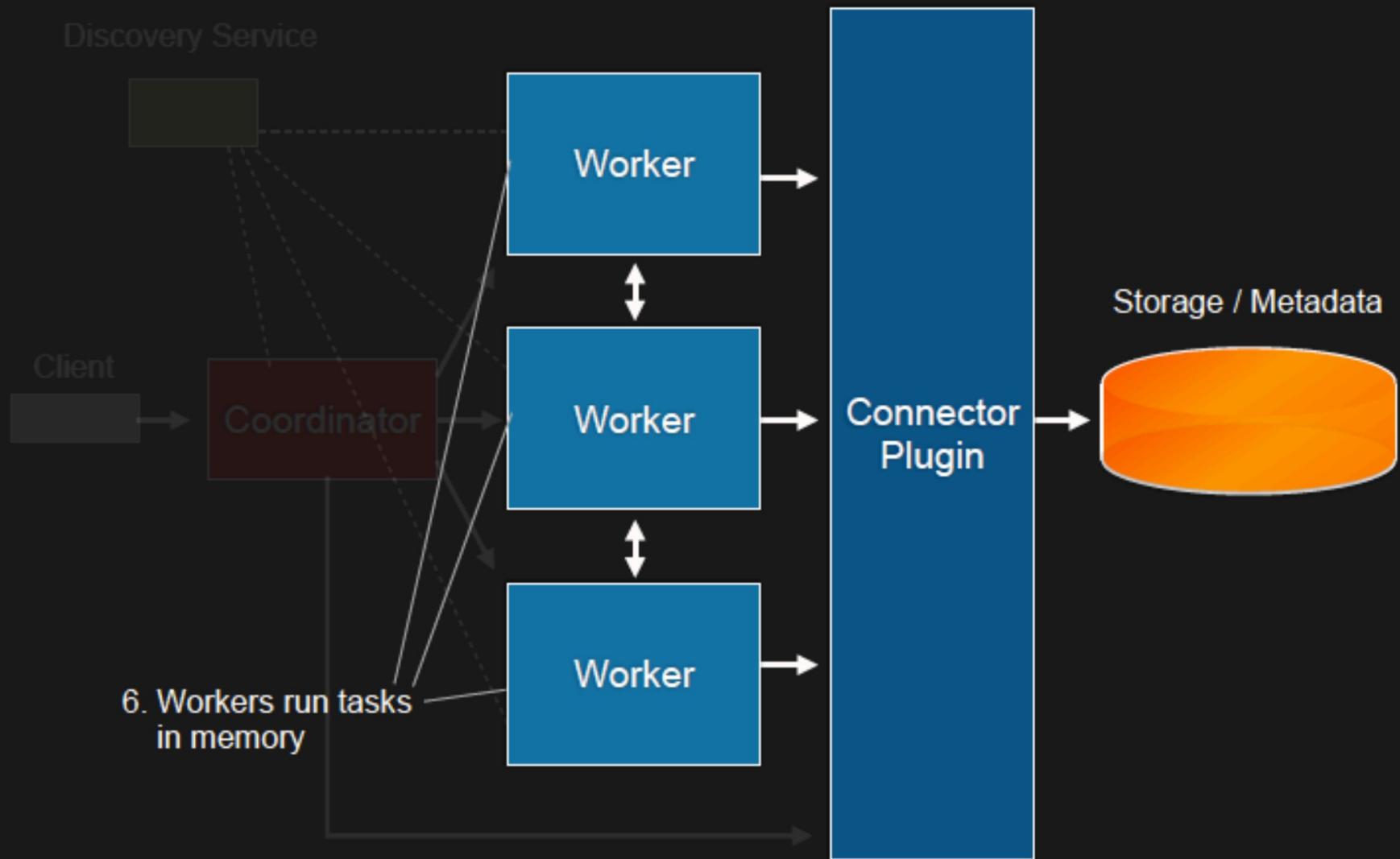


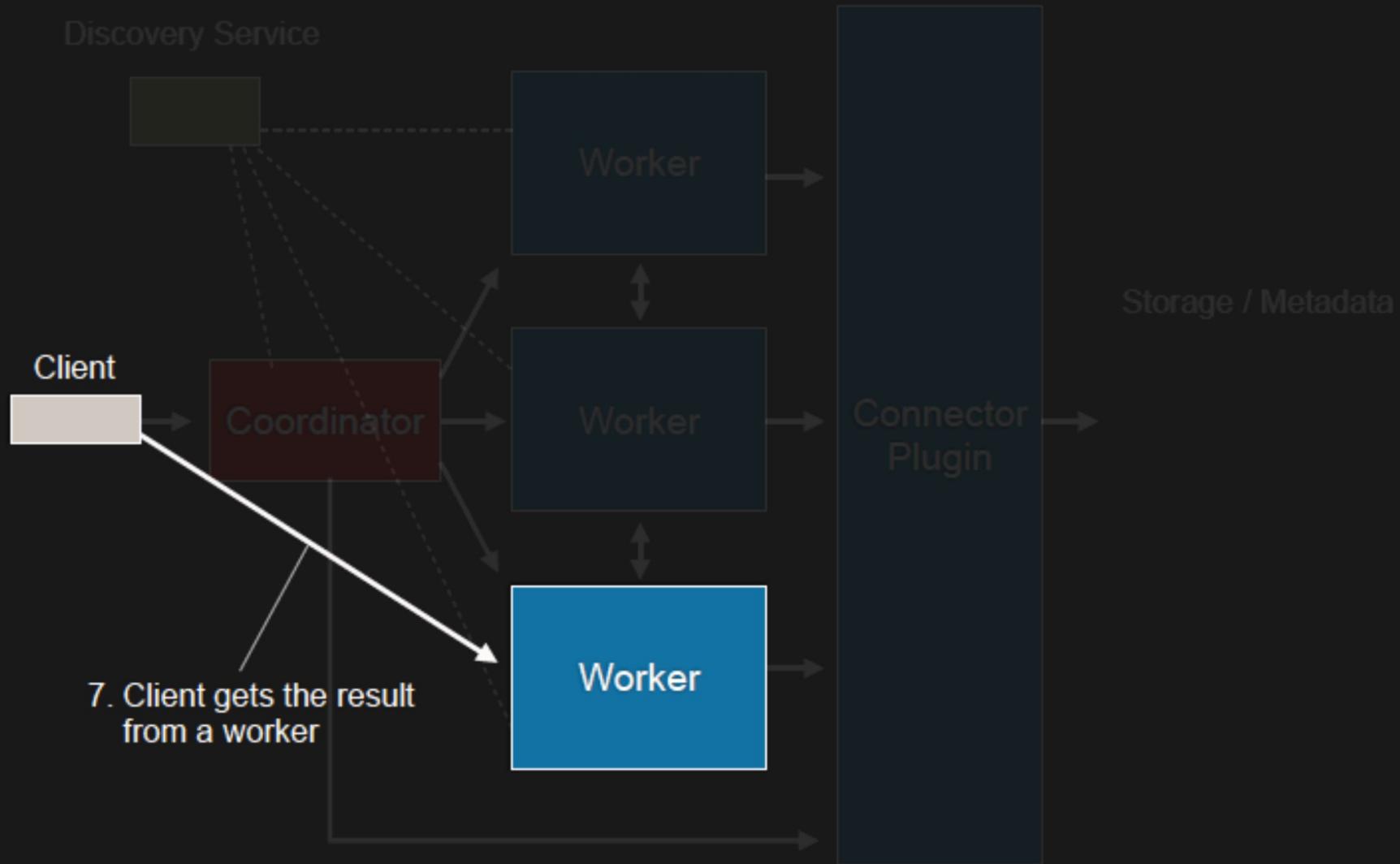


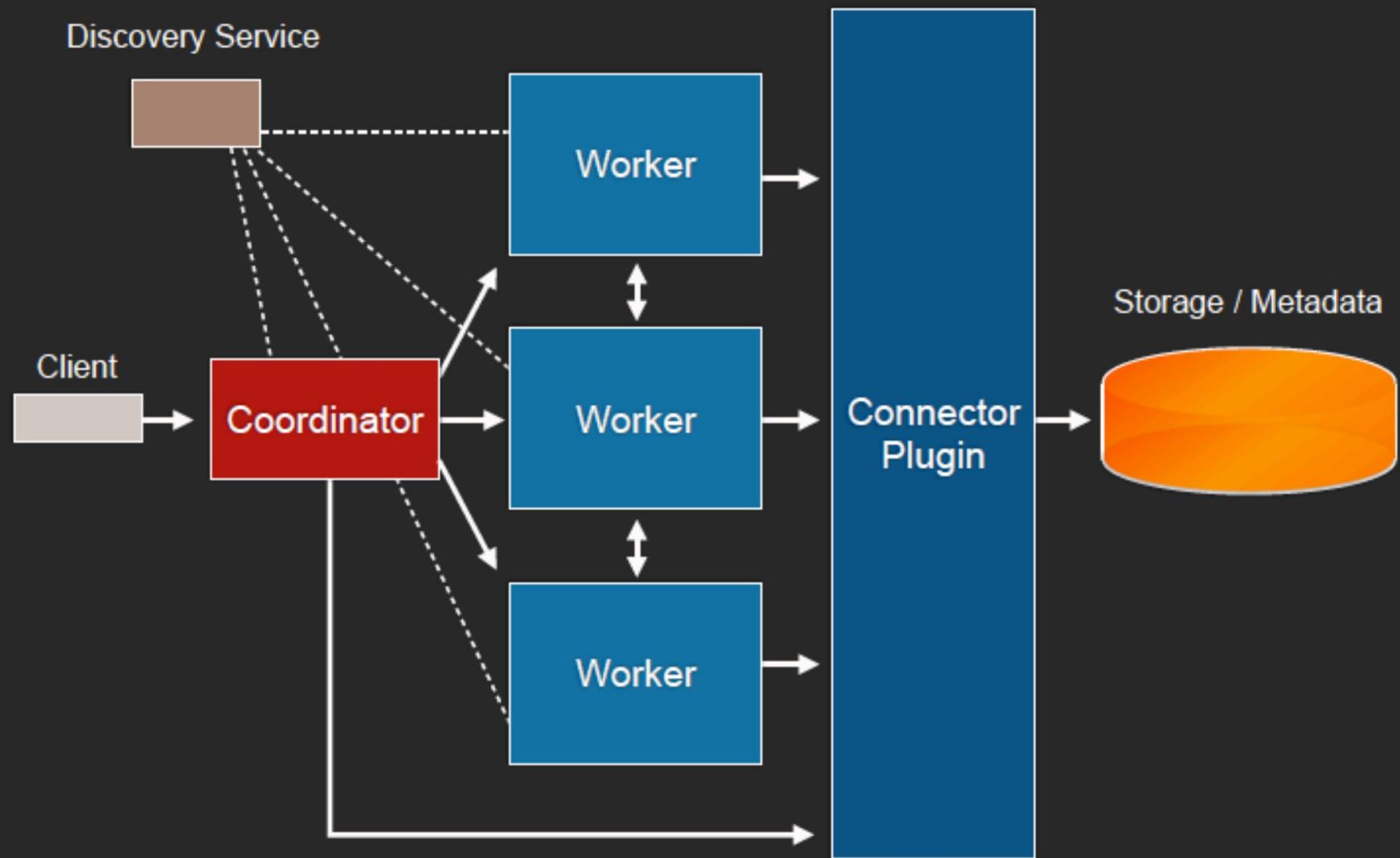








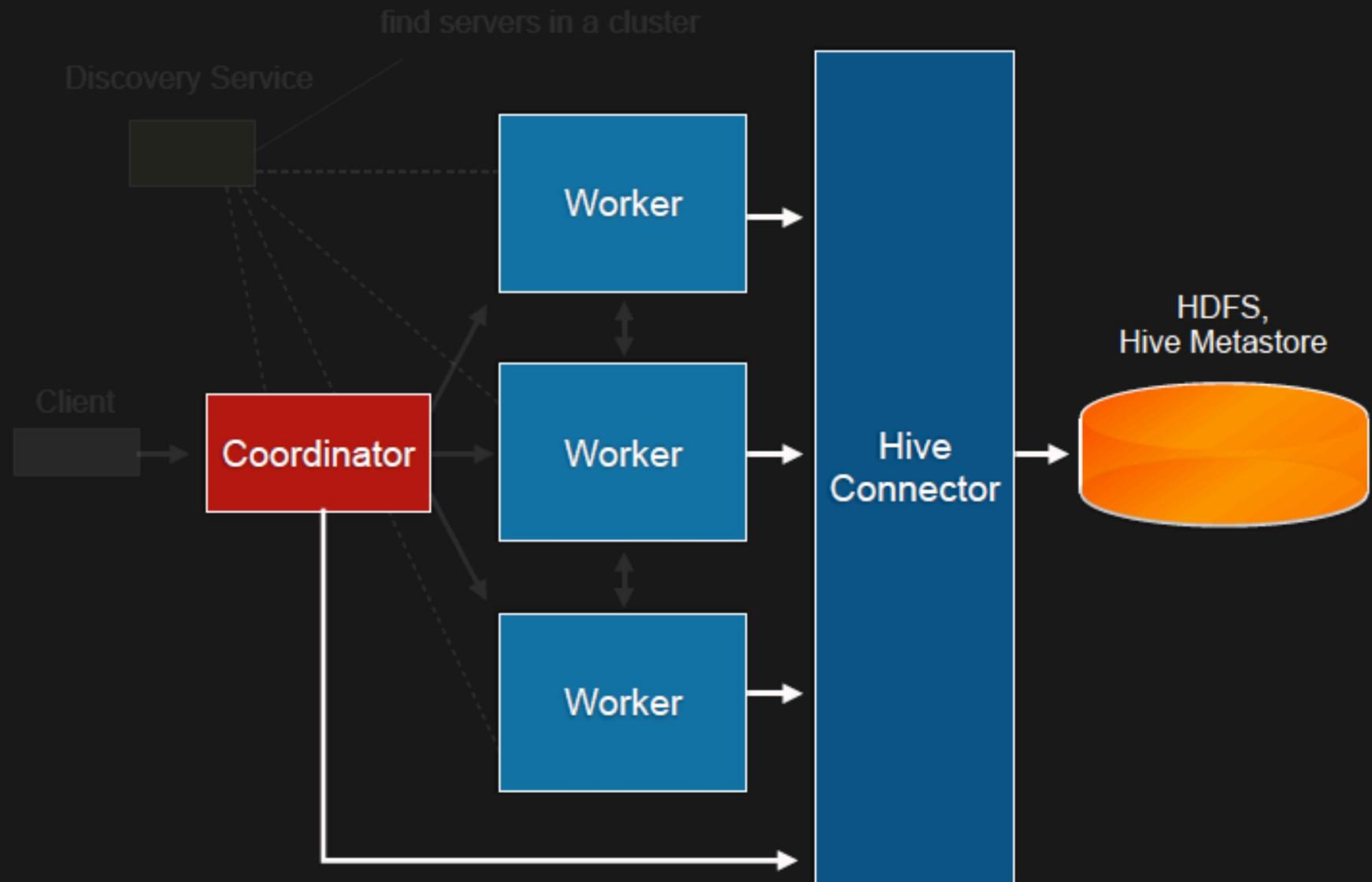




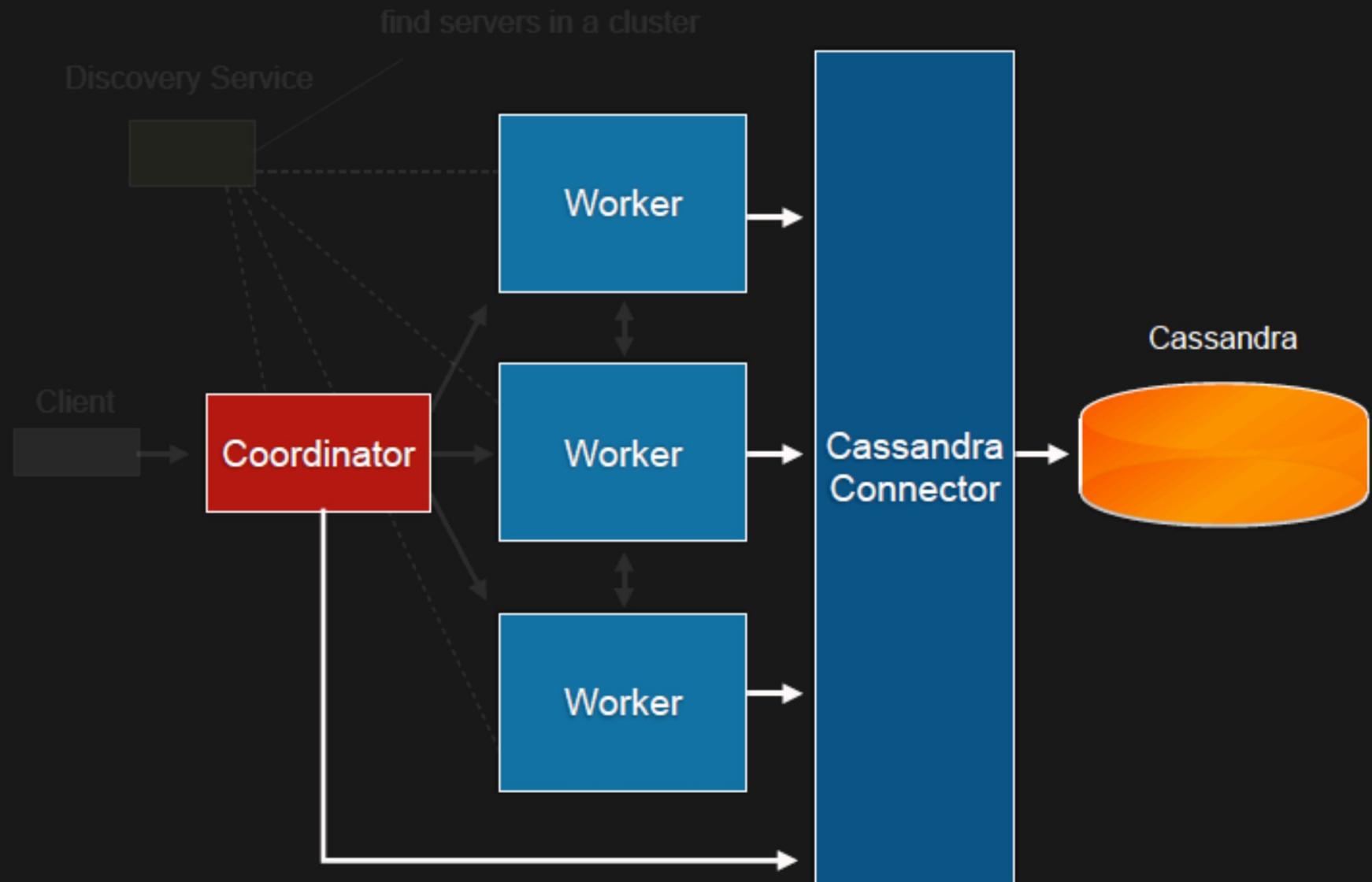
Đầu nối Presto

- Các đầu nối là các plugin cho Presto
 - được viết bằng Java
- Truy cập vào bộ nhớ và siêu dữ liệu
 - cung cấp lược đồ bảng cho người điều phối
 - cung cấp các hàng bảng cho người lao động
- Triển khai • Bộ kết nối
 - Hive • Bộ kết nối
 - Cassandra • Bộ kết nối MySQL
 - thông qua JDBC (bản phát hành trước) • Hoặc bộ kết nối của riêng bạn

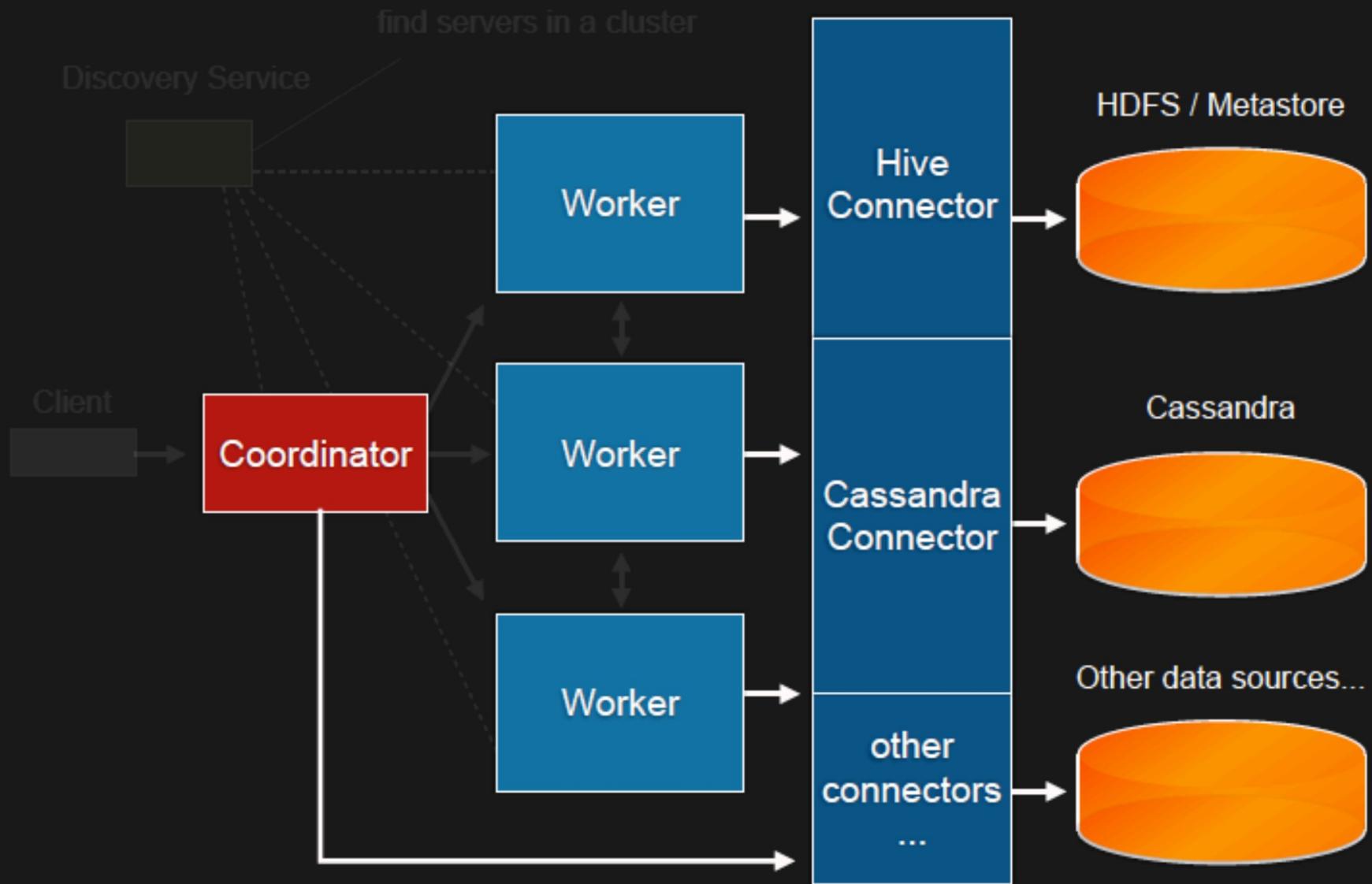
Hive connector



Cassandra connector



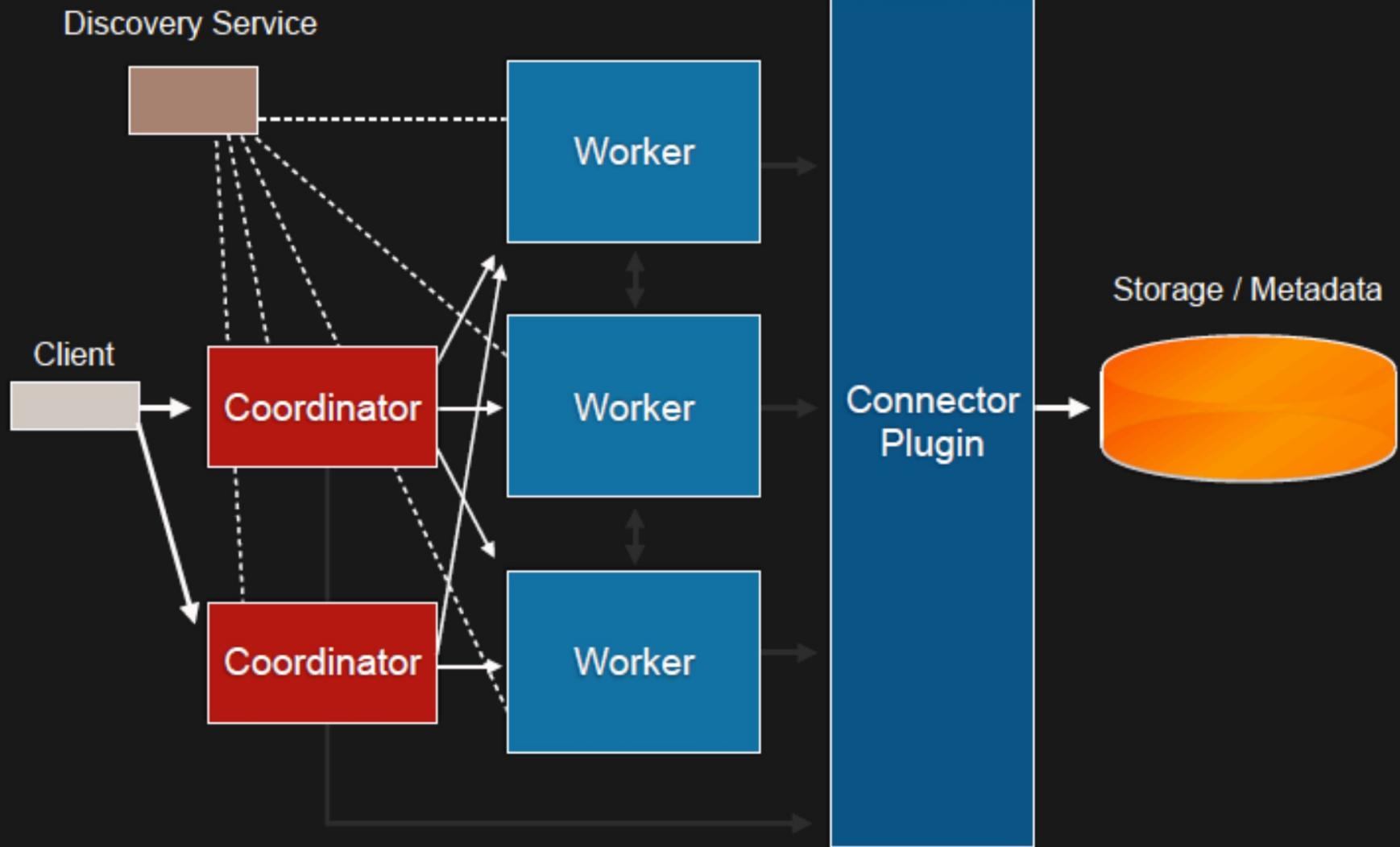
Multiple connectors in a query



Kiến trúc phân tán

- 3 loại máy chủ:
 - Điều phối viên, công nhân, dịch vụ khám phá
 - Nhận dữ liệu/siêu dữ liệu thông qua các plugin kết nối.
 - Presto KHÔNG phải là cơ sở dữ liệu
- Presto cung cấp SQL cho các kho dữ liệu hiện có
- Giao thức máy khách là HTTP + JSON •
Liên kết ngôn ngữ: Ruby, Python, PHP, Java (JDBC), R, Node.JS...

Coordinator HA

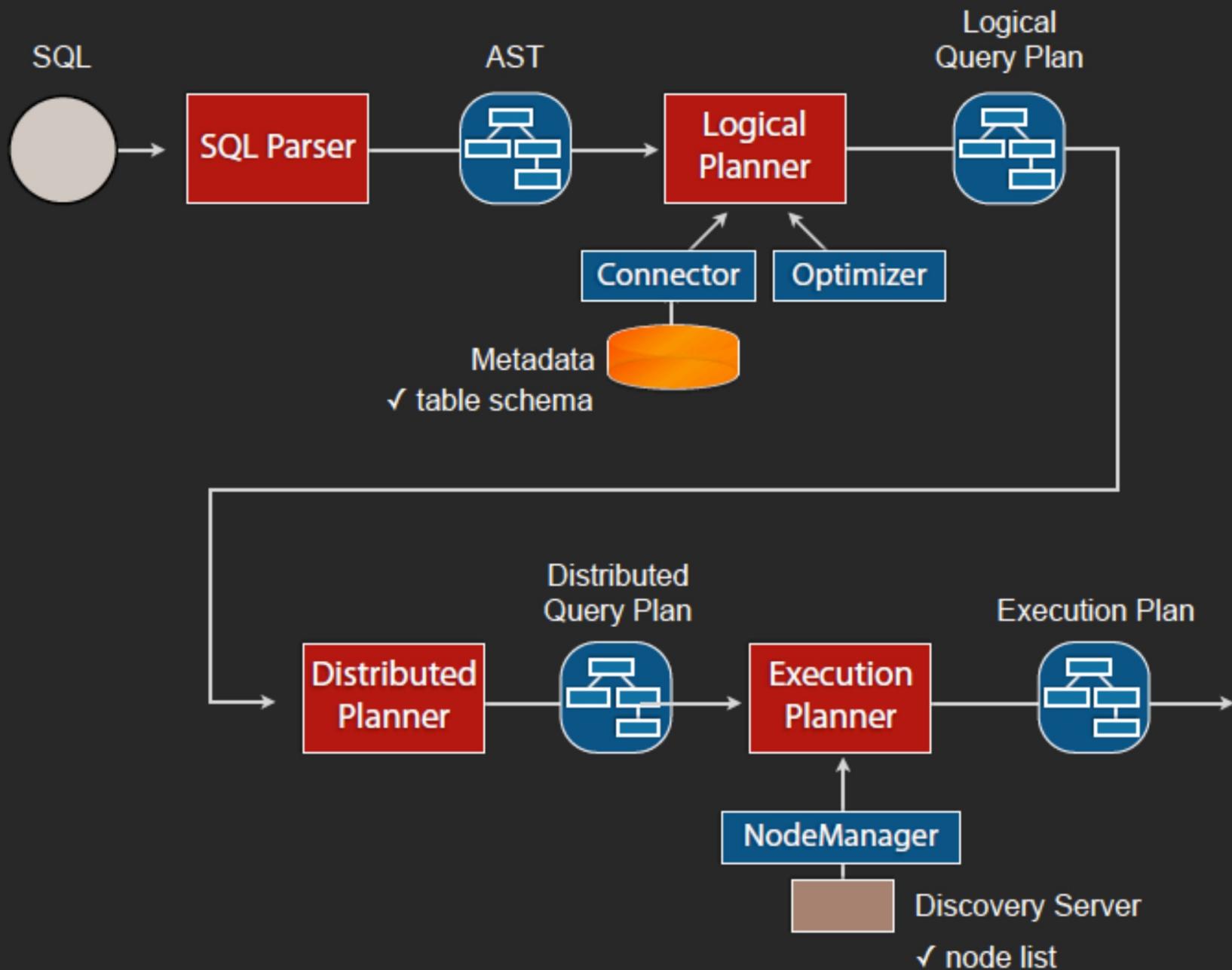


Mô hình thực hiện Presto

- Presto KHÔNG phải là MapReduce • Kế hoạch truy vấn của Presto dựa trên DAG
 - giống như Apache Tez hoặc cơ sở dữ liệu MPP truyền thống •

Truy vấn chạy như thế nào?

- Điều phối viên
 - SQL Parser
 - Query Planner
 - Execution planner
- Công nhân
 - Lập lịch thực hiện tác vụ



Trình lập kế hoạch truy vấn

Query Planner

SQL

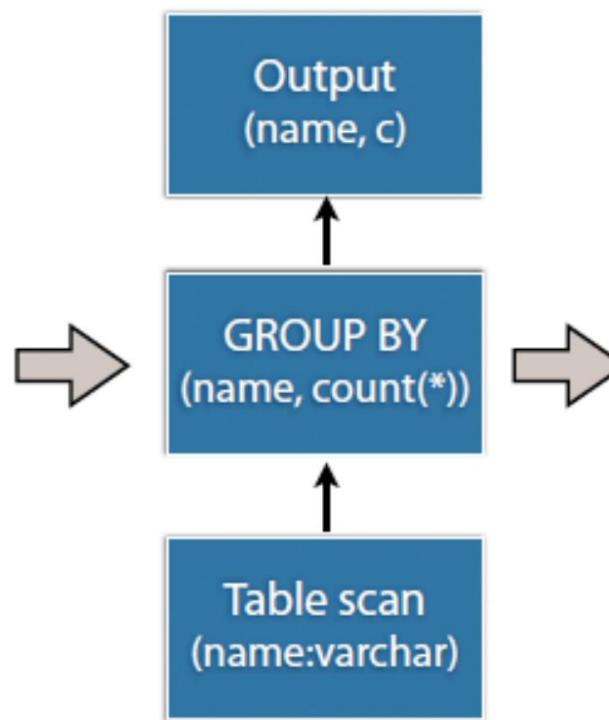
```
SELECT
    name,
    count(*) AS c
FROM impressions
GROUP BY name
```

+

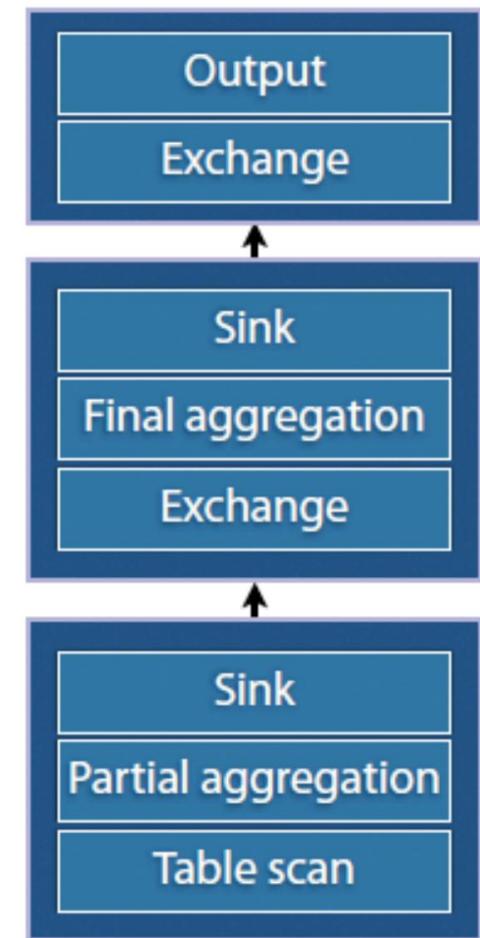
Table schema

```
impressions (
    name varchar
    time bigint
)
```

Logical query plan



Distributed query plan



Query Planner - Stages

Stage-0

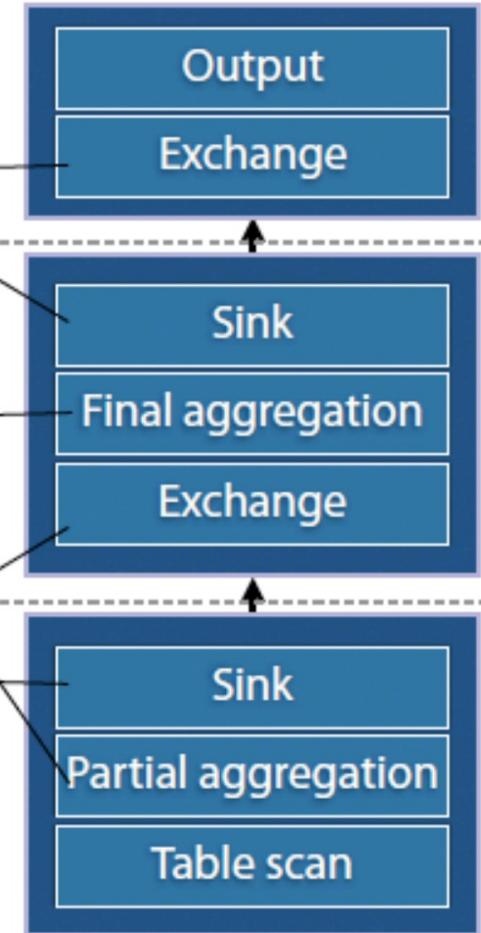
**inter-worker
data transfer**

Stage-1

**pipelined
aggregation**

Stage-2

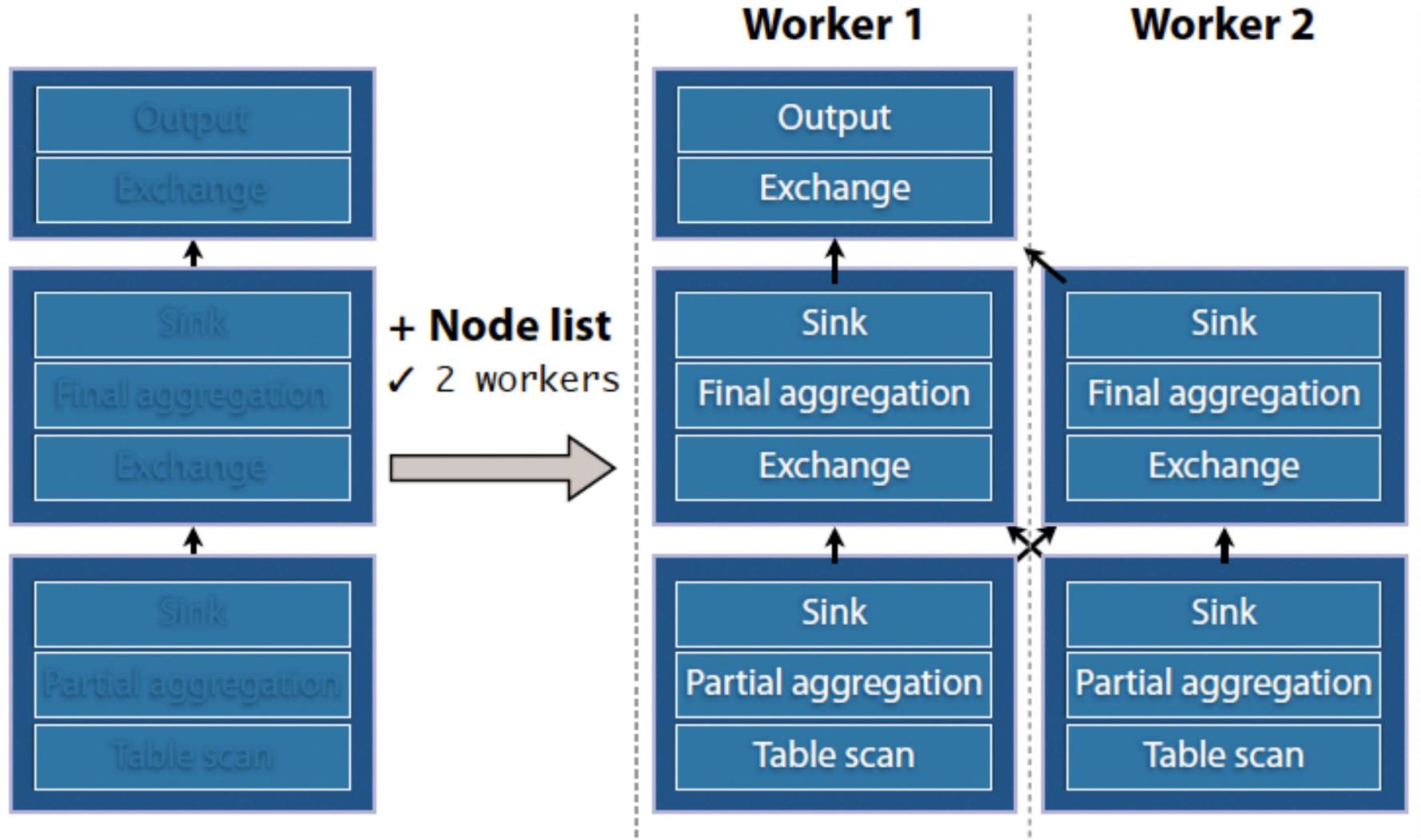
**inter-worker
data transfer**



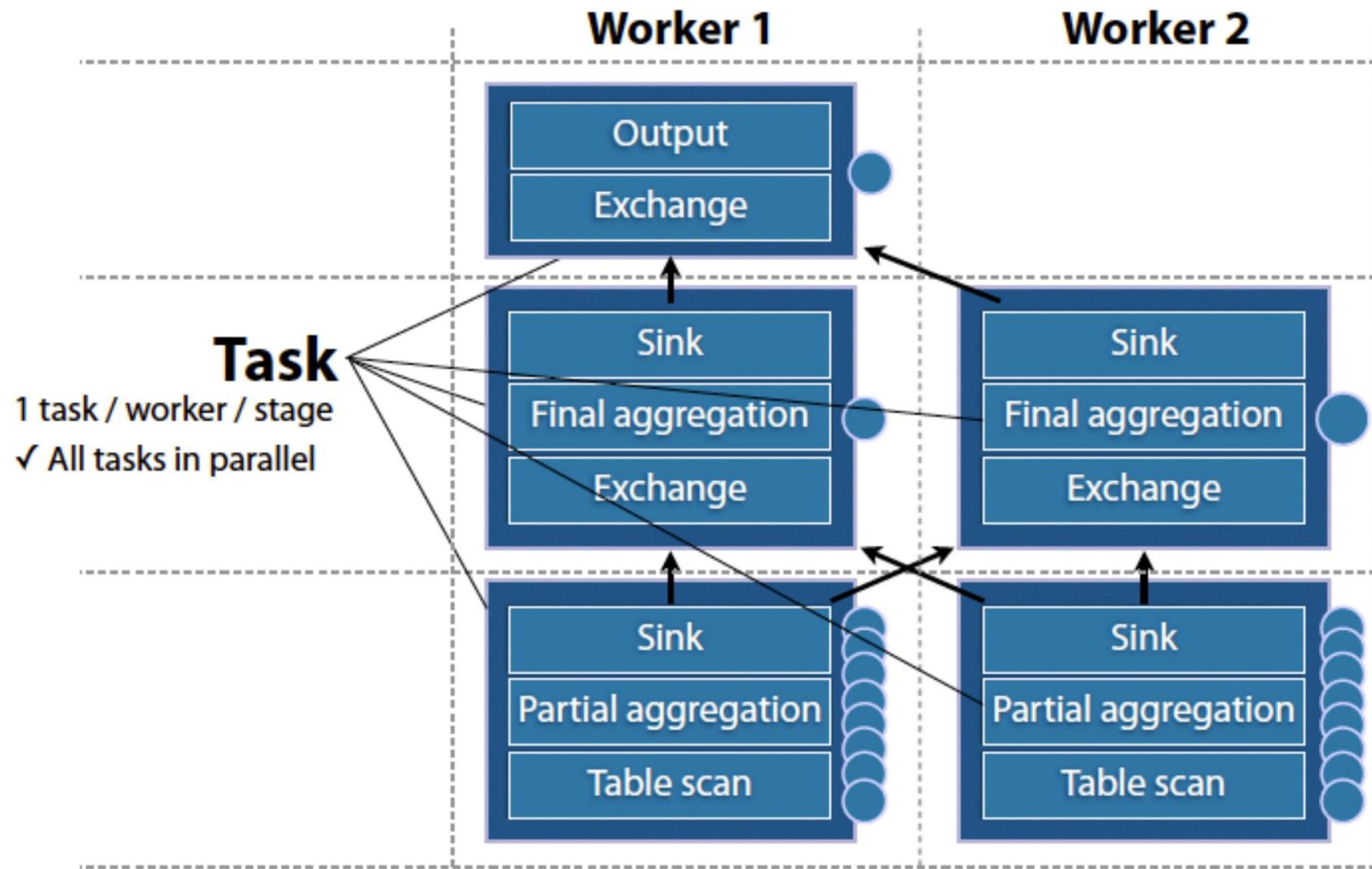
Các giai đoạn

- Giai đoạn là một phần của kế hoạch có thể được thực hiện song song giữa các công nhân
 - Công nhân thực hiện cùng một phép tính trên các tập đầu vào khác nhau dữ liệu
 - Chuyển dữ liệu đệm trong bộ nhớ (trộn) giữa các giai đoạn để cho phép trao đổi dữ liệu
- Việc xáo trộn làm tăng độ trễ, sử dụng hết bộ nhớ đệm và có chi phí CPU cao

Execution Planner

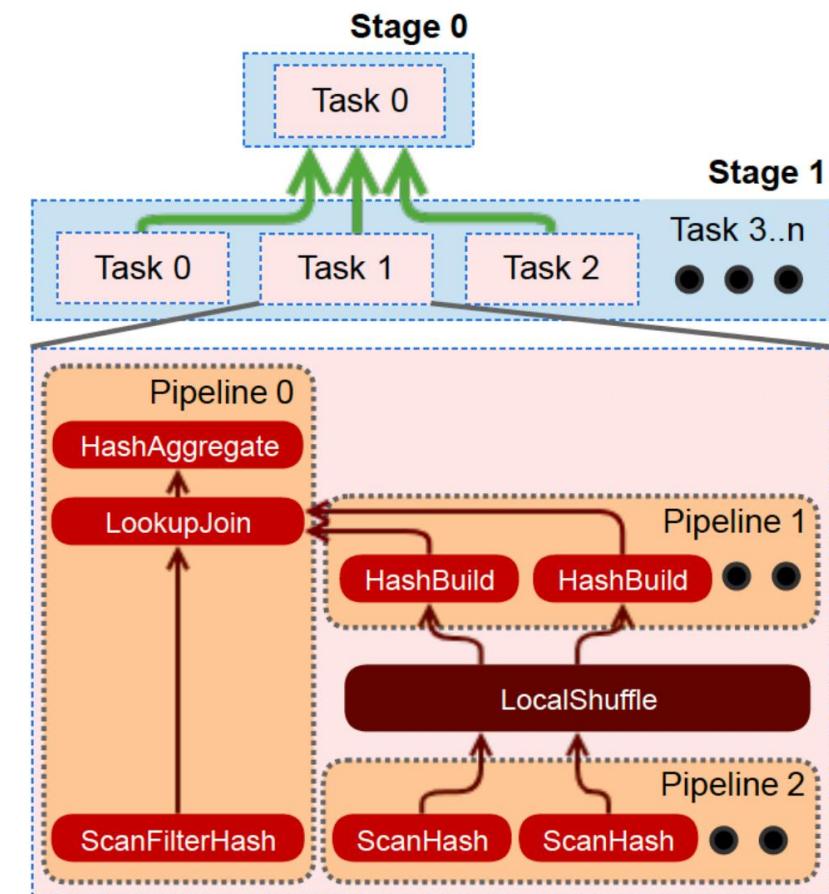


Execution Planner - Tasks



Nhiệm vụ

- Người điều phối phân phối
lập kế hoạch các giai đoạn cho công nhân dưới
dạng các nhiệm vụ có thể thực hiện được
- 1 nhiệm vụ / 1 công nhân / 1 giai đoạn
- Một tác vụ có thể có nhiều đường ống
- Một đường ống bao gồm một chuỗi các nhà
điều hành



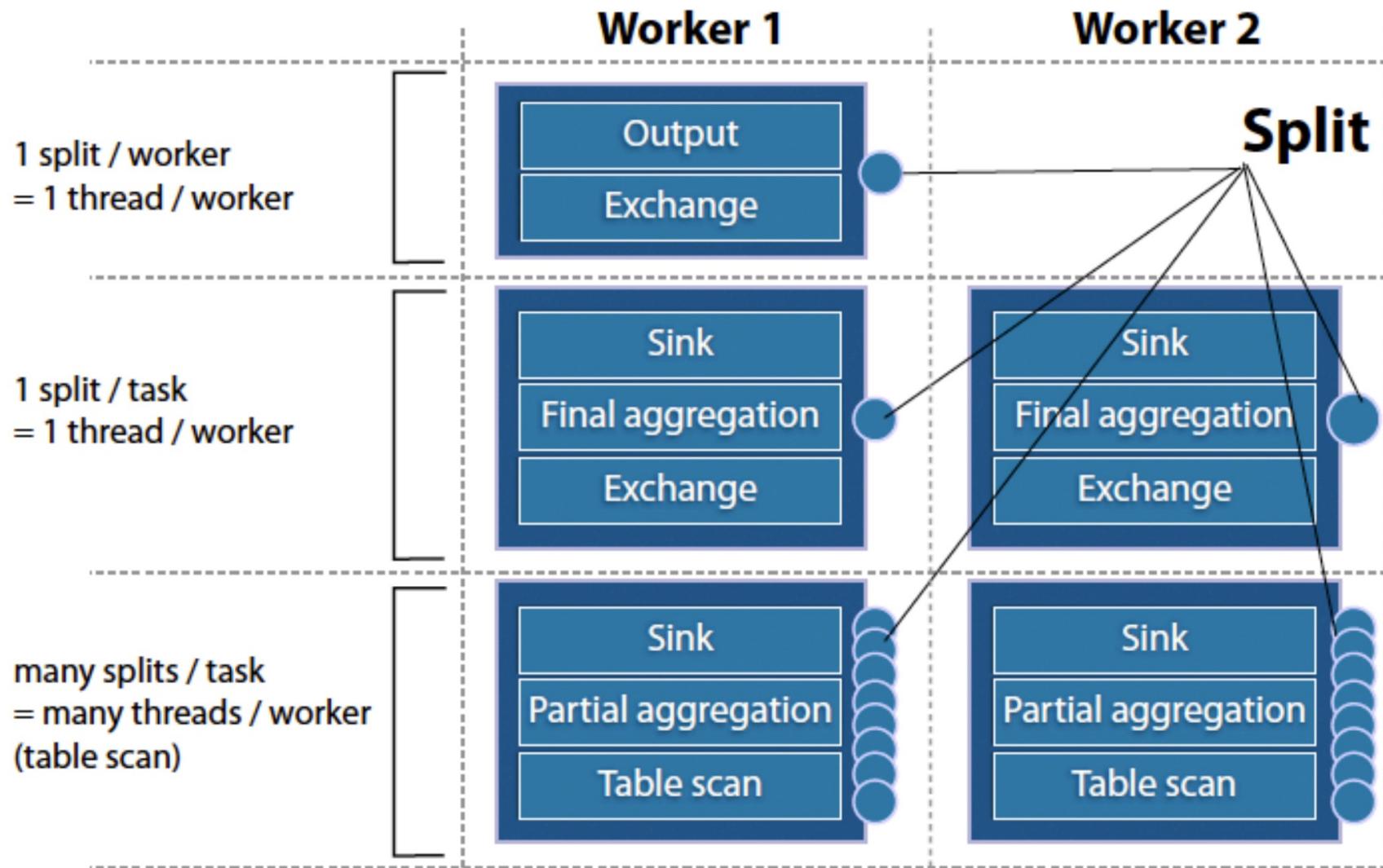
Lên lịch

- Để thực hiện một truy vấn, Scheduler đưa ra hai bộ quyết định lập lịch
- Lên lịch sân khấu
 - tất cả cùng một lúc: giảm thiểu thời gian làm việc bằng cách lên lịch tất cả các giai đoạn thực hiện đồng thời
 - dữ liệu được xử lý ngay khi có sẵn
 - lợi ích cho các trường hợp sử dụng nhạy cảm với độ trễ như Tương tác Phân tích, Phân tích nhà phát triển/nhà quảng cáo và Kiểm tra A/B
 - Theo giai đoạn:
 - Khi một giai đoạn được lên lịch theo chính sách, nó bắt đầu chỉ định các nhiệm vụ cho giai đoạn đó cho các nút công nhân
 - cải thiện hiệu quả bộ nhớ cho trường hợp sử dụng Phân tích hàng loạt

Lên lịch nhiệm vụ

- Các giai đoạn được phân loại thành giai đoạn lá và giai đoạn trung gian
- Giai đoạn lá
 - trình lập lịch tác vụ sẽ tính đến các ràng buộc do mạng và trình kết nối áp đặt khi chỉ định tác vụ cho các nút công nhân
 - công nhân cùng vị trí và các nút lưu trữ
 - Ràng buộc bố trí dữ liệu kết nối
- Giai đoạn trung gian
 - Nhiệm vụ cho các giai đoạn trung gian có thể được giao cho bất kỳ công nhân nào nút
 - Động cơ vẫn cần phải quyết định có bao nhiêu tác vụ sẽ được lên lịch cho từng giai đoạn

Execution Planner - Split



Chia lịch trình

- Các phần chia tách là các tay cầm mờ đục cho một khối có thể định địa chỉ dữ liệu
 - trong hệ thống lưu trữ bên ngoài
 - hoặc kết quả trung gian do những người lao động khác tạo ra
 - Ví dụ. Đọc từ HDFS
 - Một split là một đường dẫn tệp và bù trừ vào một vùng của tệp
- Phân chia nhiệm vụ
 - Phân chia được gán cho từng nhiệm vụ một cách chậm rãnh
 - Các phần chia được liệt kê khi truy vấn thực thi, không phải trước
 - Các truy vấn có thể bắt đầu tạo ra kết quả mà không cần xử lý tất cả dữ liệu
 - Các phần chia được gán cho công nhân có hàng đợi ngắn nhất
 - Giảm mức sử dụng bộ nhớ siêu dữ liệu trên bộ điều phối

Tối ưu hóa truy vấn: Bố cục dữ liệu

- Optimizer tận dụng lợi thế của bố cục vật lý của dữ liệu • Thuộc tính: phân vùng, sắp xếp, nhóm, chỉ mục • Bảng có thể có nhiều bố cục với các thuộc tính khác nhau • Bố cục có thể có một tập hợp con các cột hoặc dữ liệu
- Trình tối ưu hóa chọn bố cục tốt nhất cho truy vấn • Điều chỉnh truy vấn bằng cách thêm bố cục vật lý mới

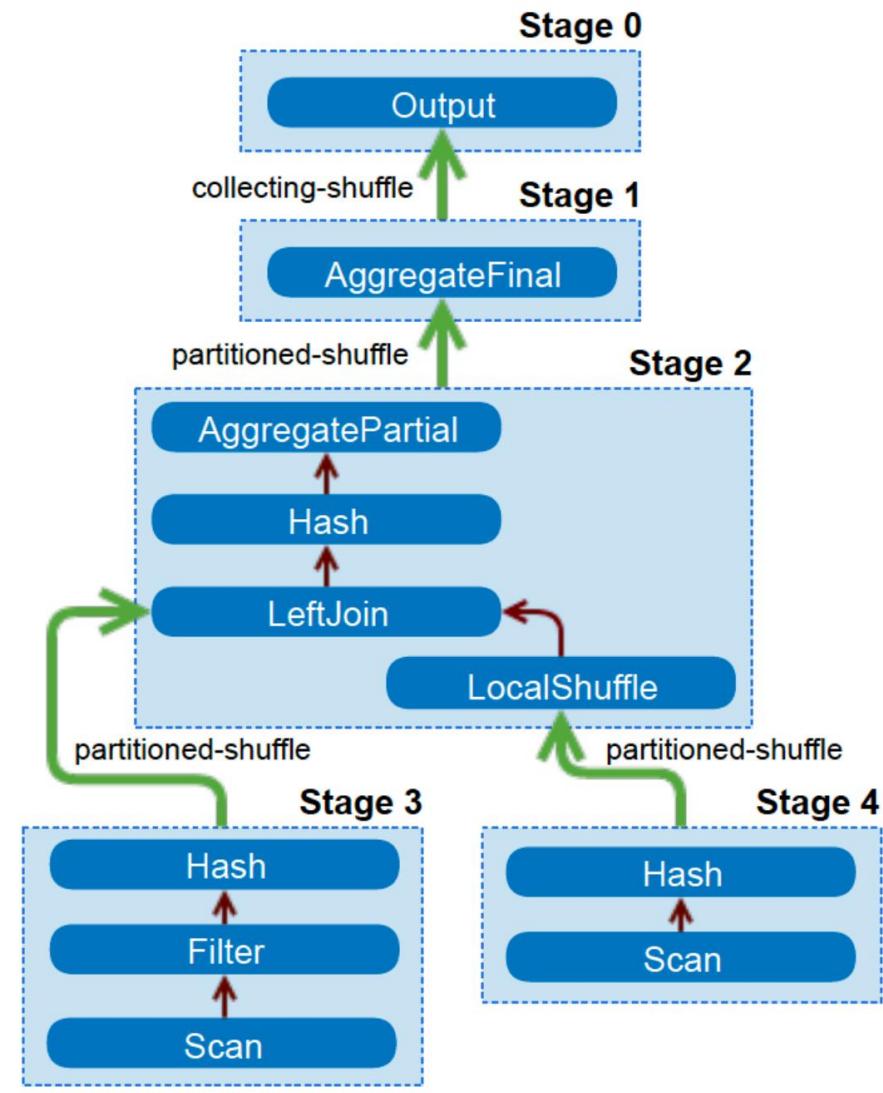
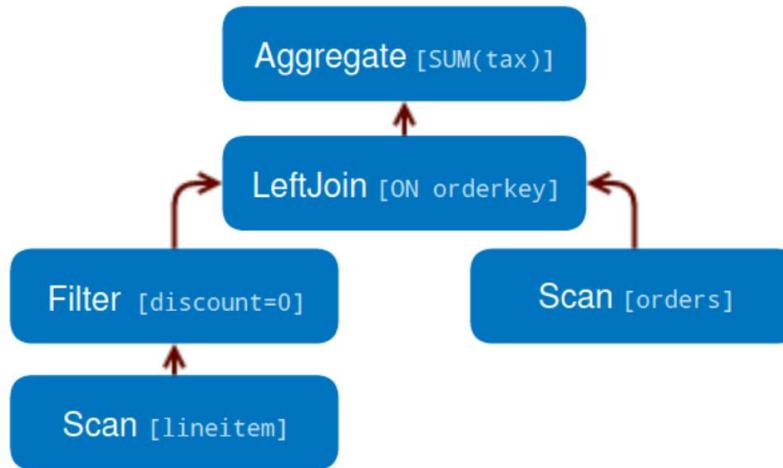
Tối ưu hóa truy vấn: Đẩy xuống vị ngữ

- Bộ tối ưu hóa có thể đẩy các vị từ phạm vi và bình đẳng xuống thông qua trình kết nối để cải thiện hiệu quả lọc
- Engine cung cấp các đầu nối có ràng buộc hai phần:
 - Miền giá trị: phạm vi và giá trị null
 - Vị ngữ “hộp đen” để lọc

Ví dụ.

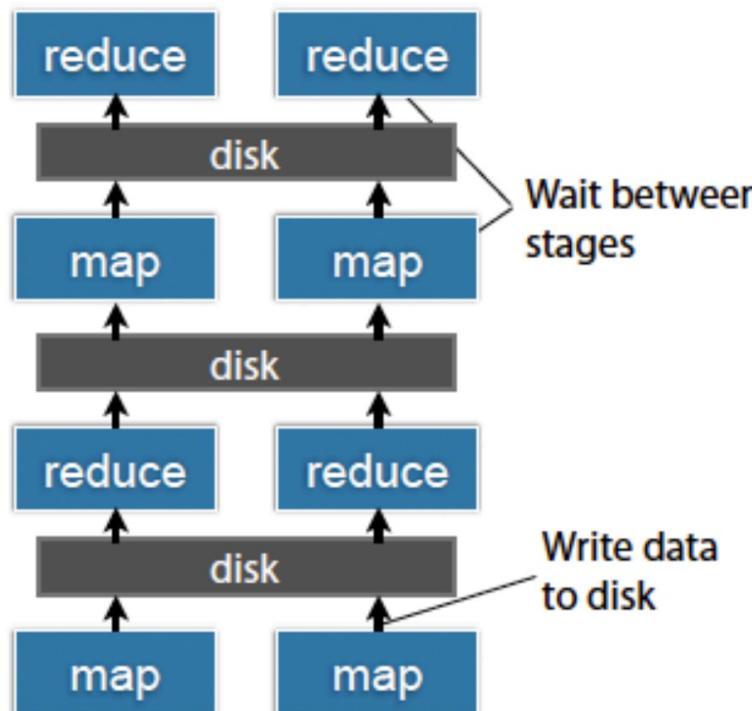
```

SELECT
    orders.orderkey, SUM(tax)
FROM orders
LEFT JOIN lineitem
    ON orders.orderkey = lineitem.orderkey
WHERE discount = 0
GROUP BY orders.orderkey
  
```

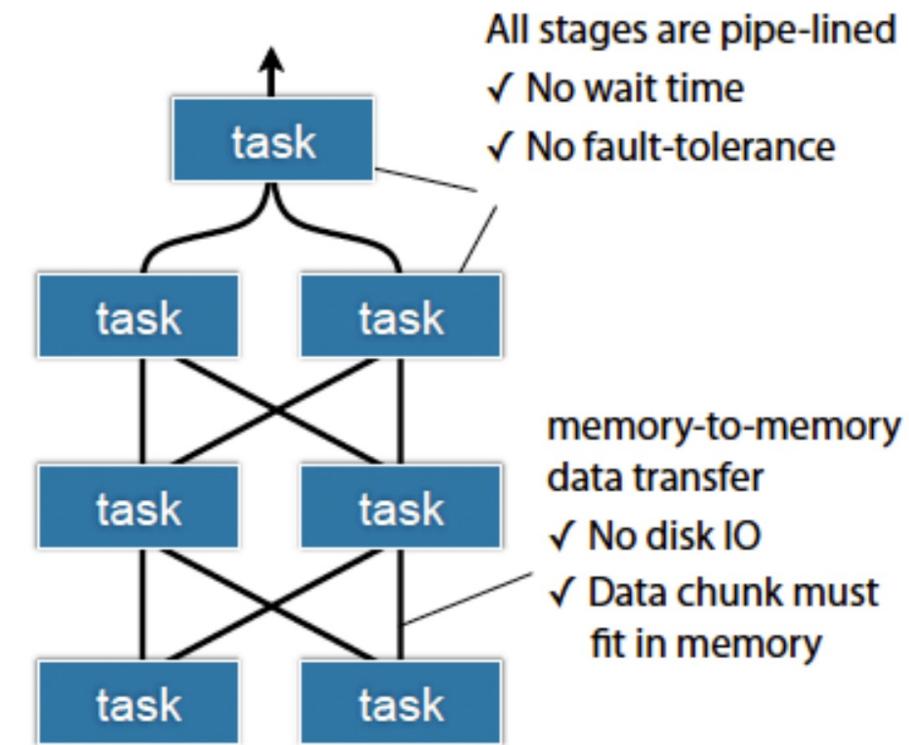


MapReduce vs. Presto

MapReduce



Presto



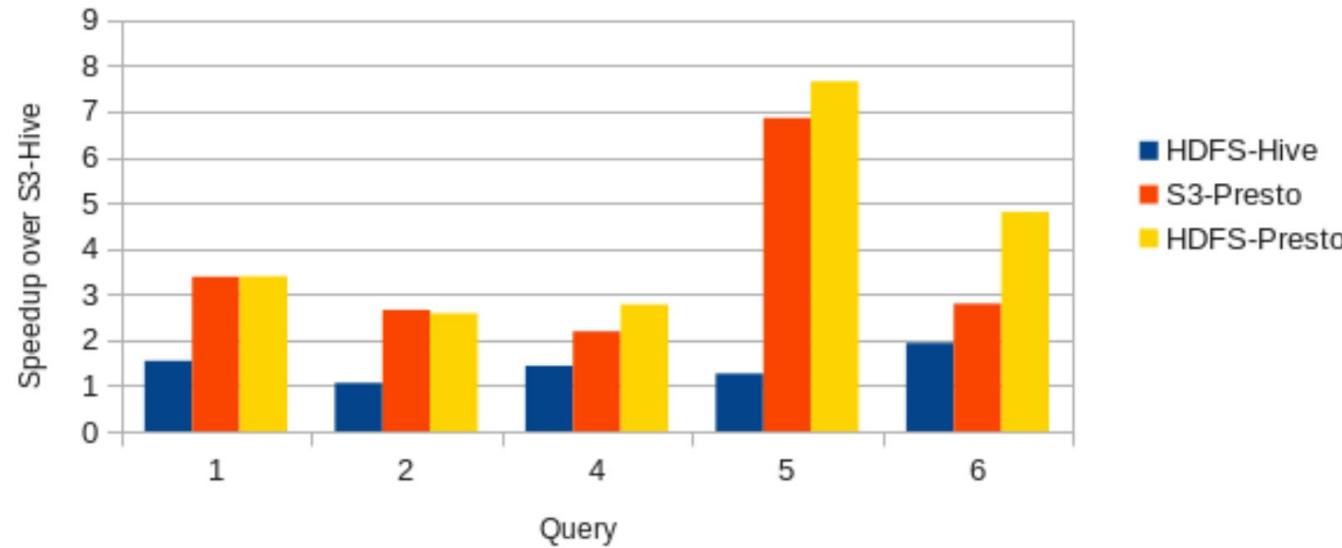
Thực hiện truy vấn

- SQL được chuyển đổi thành các giai đoạn, tác vụ và phân tách • Tất cả các tác vụ chạy song song • Không có thời gian chờ giữa các giai đoạn (theo đường ông) • Nếu một tác vụ bị lỗi, tất cả các tác vụ đều bị lỗi cùng lúc (truy vấn bị lỗi) • Truyền dữ liệu từ bộ nhớ này sang bộ nhớ khác
 - Không có IO
- đĩa • Nếu dữ liệu tổng hợp không vừa với bộ nhớ, truy vấn sẽ không thành công • Lưu ý: truy vấn chết nhưng công nhân không chết. Việc sử dụng bộ nhớ của tất cả các truy vấn được quản lý hoàn toàn

Điểm chuẩn Qubole Presto

TPC-H* Comparison

75GB, RCFile, 10 m1.xlarge, Speedup over S3-Hive

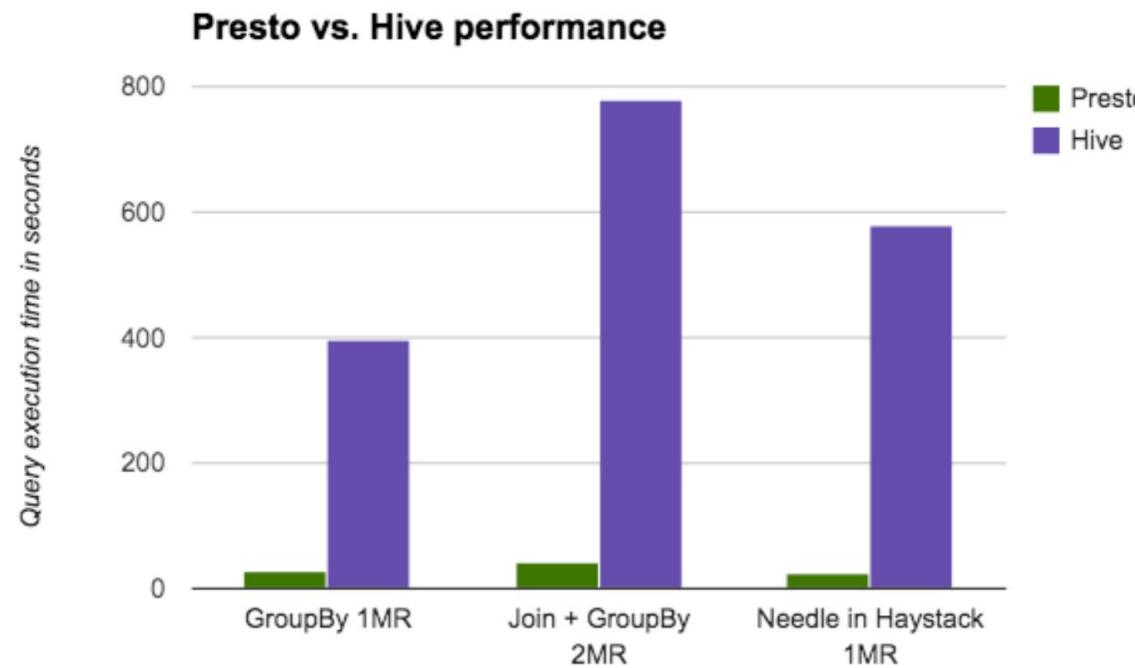


Điểm chuẩn Netflix Presto (1)

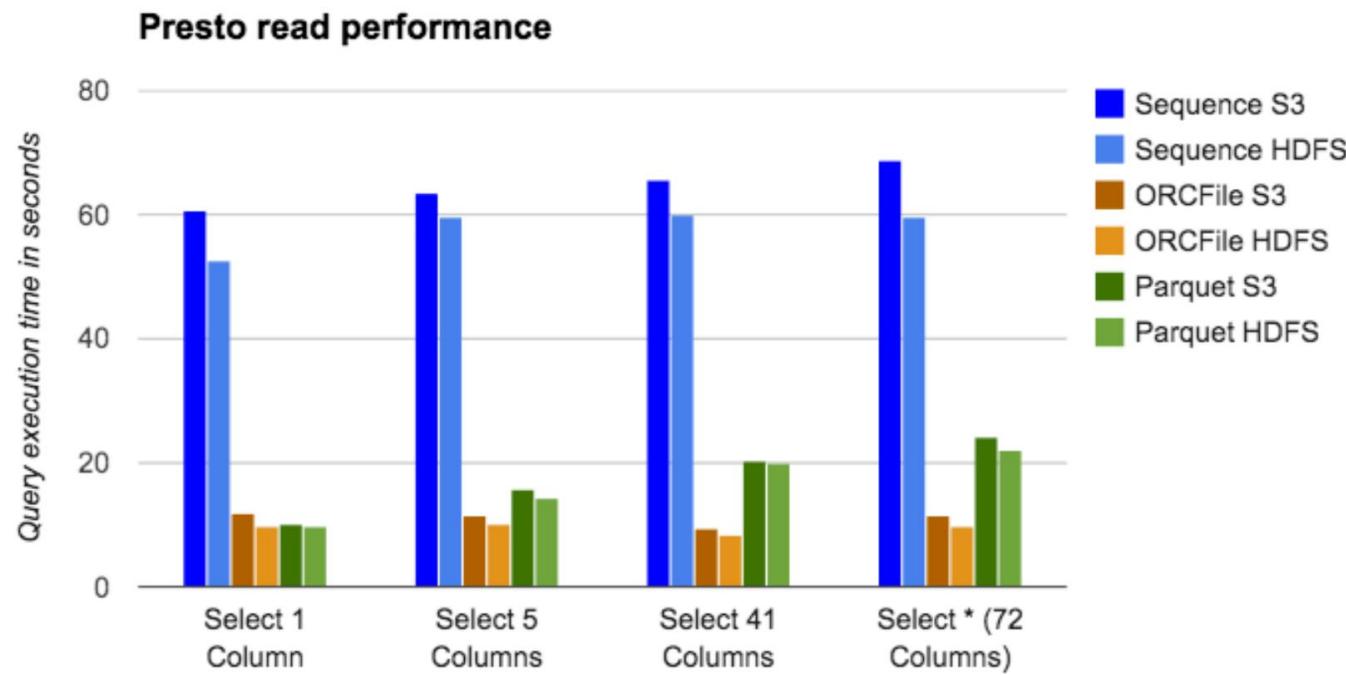
- truy vấn nhóm theo, truy vấn nối cộng với truy vấn nhóm theo và tìm kim đáy bể (quét bảng) •

Tệp đầu vào Parquet trên S3/kích thước tệp từ 140GB đến 210GB

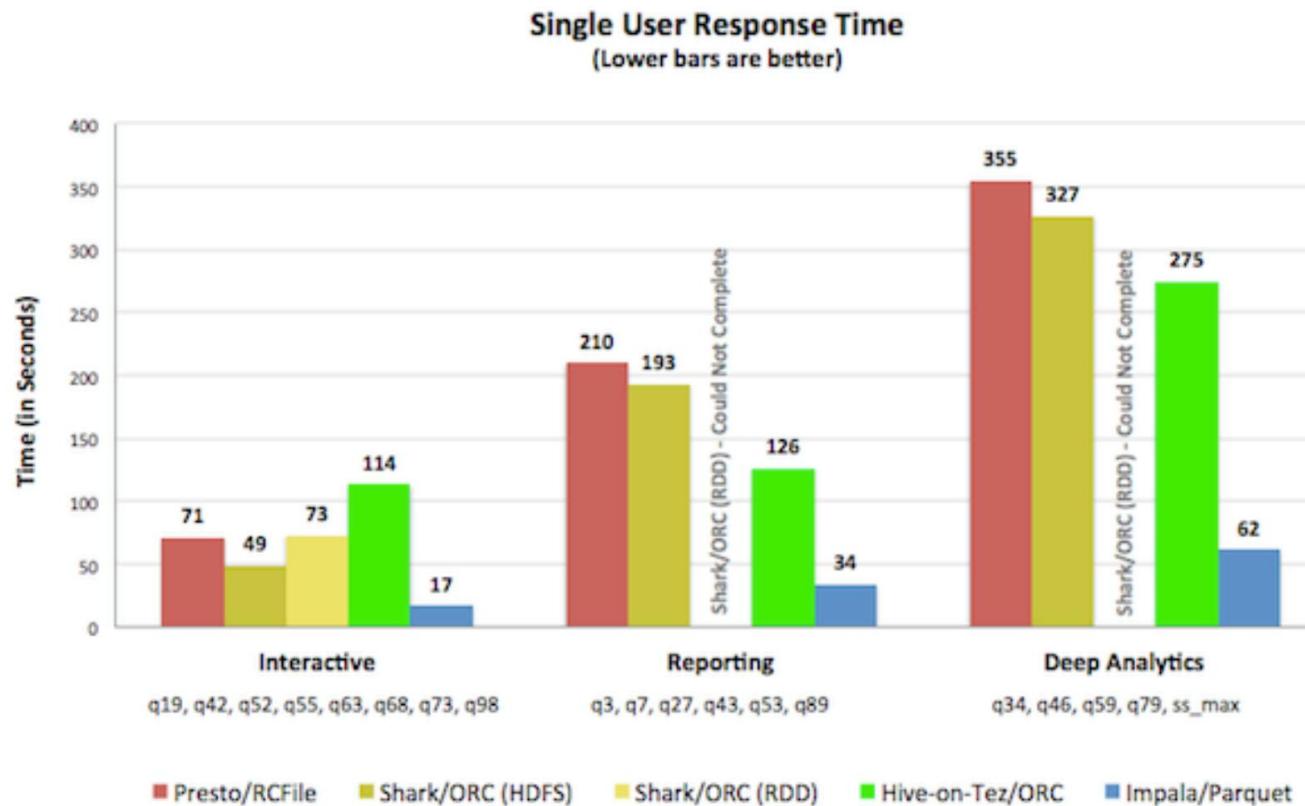
- 40 nút m2.4xlarge



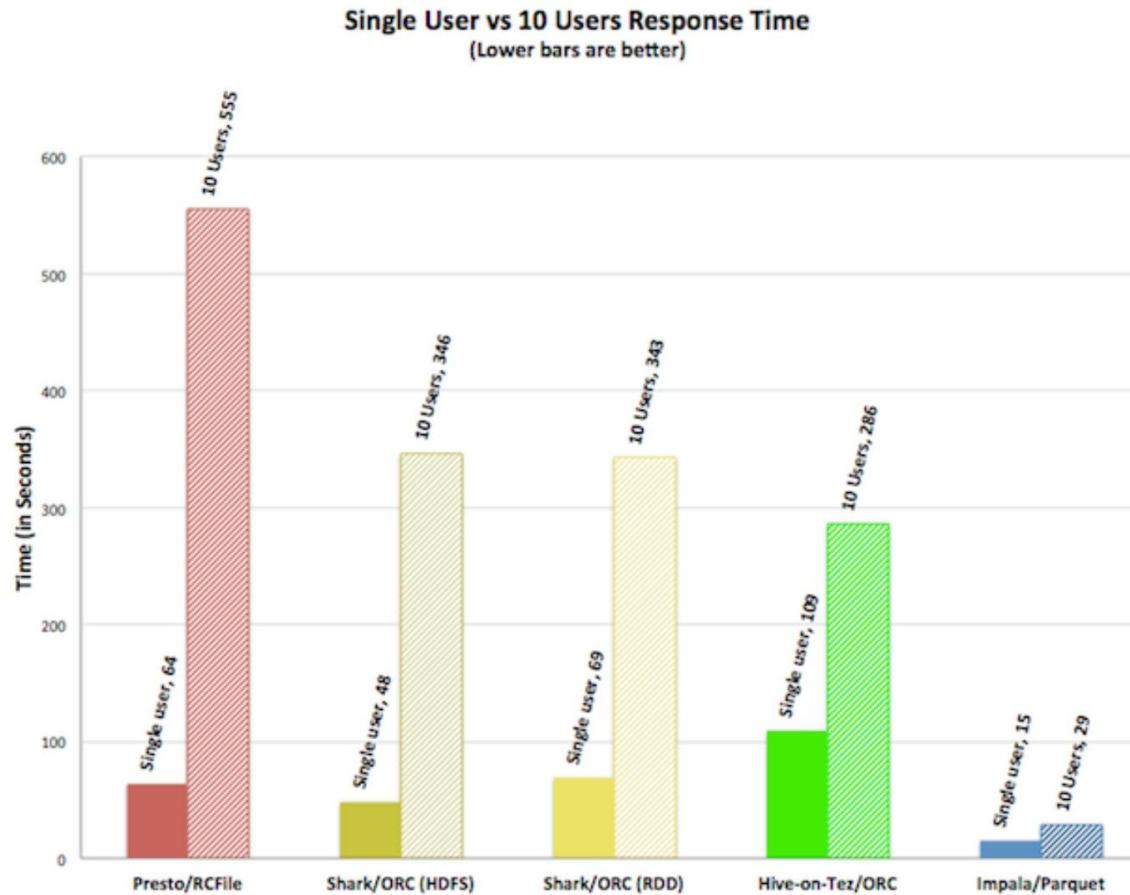
Điểm chuẩn Netflix Presto (2)



Điểm chuẩn Cloudera (1)

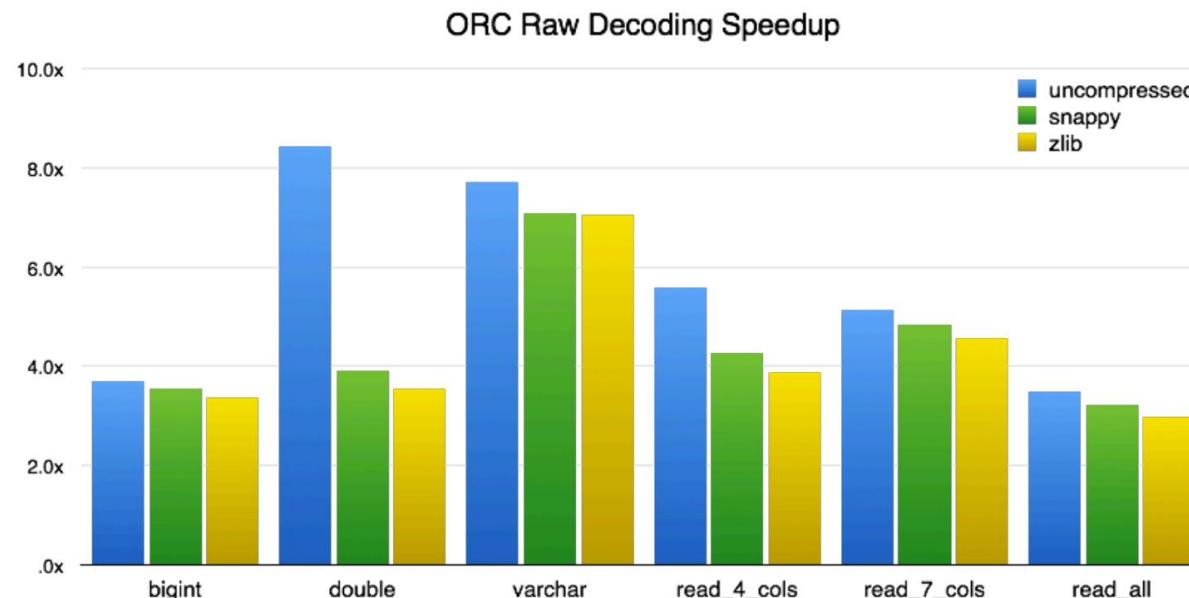


Điểm chuẩn Cloudera (2)



Thậm chí còn nhanh hơn: Dữ liệu với tốc độ của Presto ORC

- 6 triệu hàng sử dụng TPC-H • Đọc nhiều tập hợp cột khác nhau
- Đầu đọc ORC dựa trên Hive cũ so với ORC Presto mới người đọc



Tài liệu tham khảo

- <https://code.facebook.com/posts/370832626374903/en-faster-data-at-the-speed-of-presto-orc/>
- <https://www.facebook.com/notes/facebook-kỹ-thuật/presto-tương-tác-với-petabyte-dữ-liệu-tại-facebook/10151786197628920/>
- Traverso, Martin. "Presto: Tương tác với petabyte dữ liệu tại Facebook." 2014.
- <https://www.slideshare.net/frsyuki/presto-hadoop-conference-japan-2014>



25
YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Cảm ơn sự
chú ý
của bạn!!!

