

25 YEARS ANNIVERSARY
SICT

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

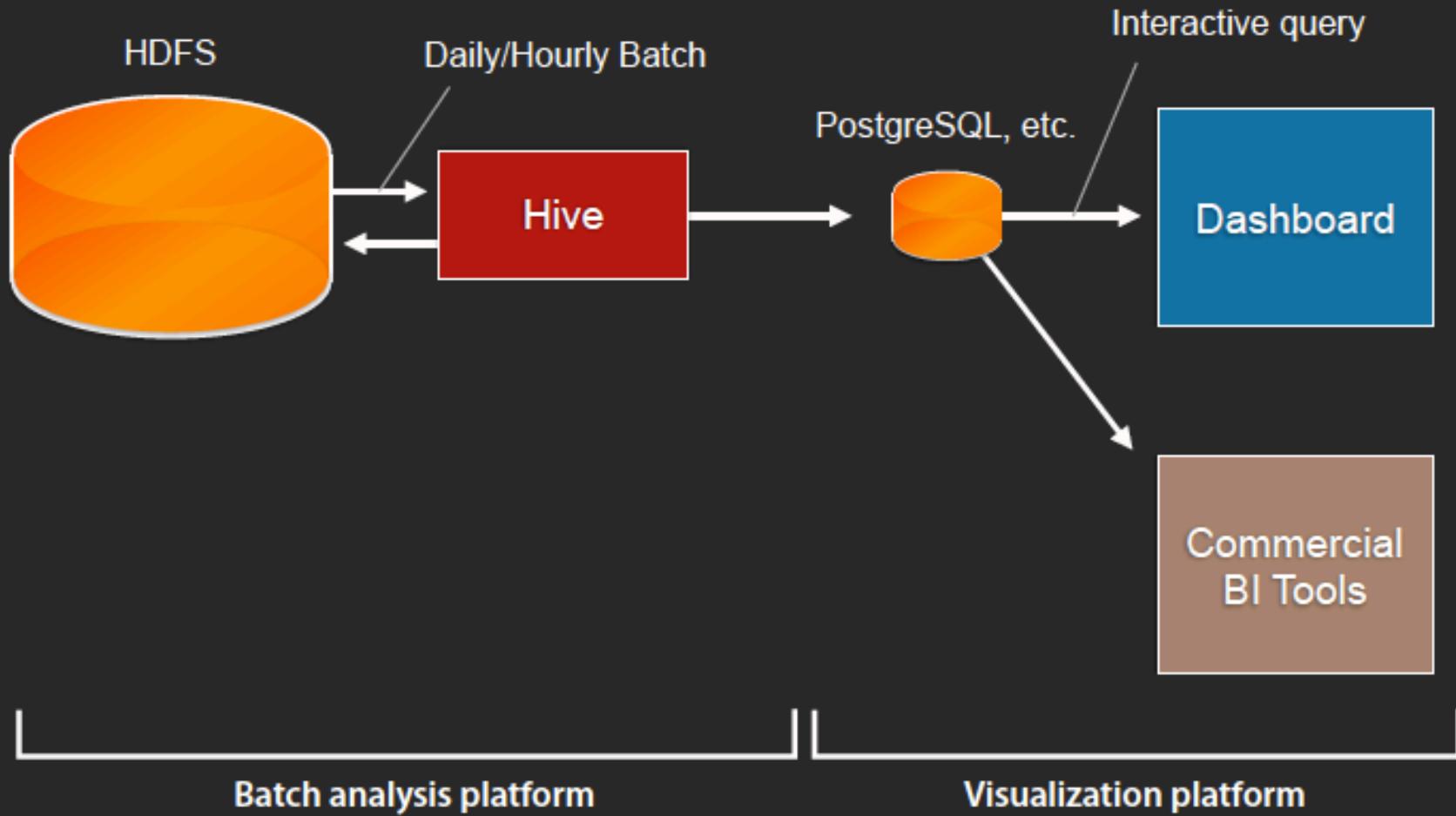
Lịch sử

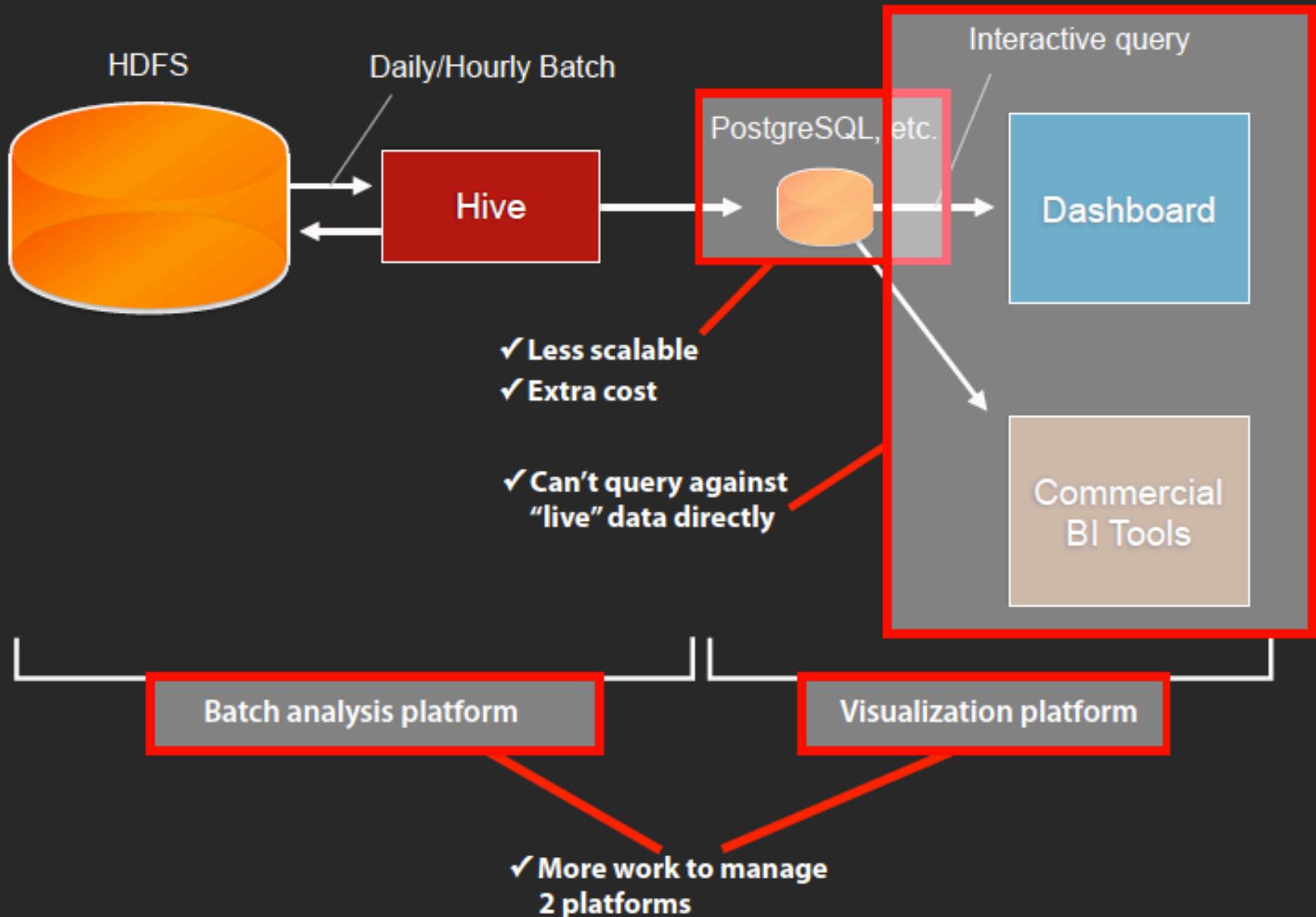
- Mùa thu 2012: Dự án bắt đầu tại Facebook
 - Được thiết kế cho truy vấn tương tác
 - với tốc độ của kho dữ liệu thương mại
 - và khả năng mở rộng theo quy mô của Facebook
- Mùa đông 2013: Nguồn mở
 - Hơn 30 người đóng góp trong 6 tháng
 - bao gồm cả những người bên ngoài Facebook
- 2019: Hơn 300 người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

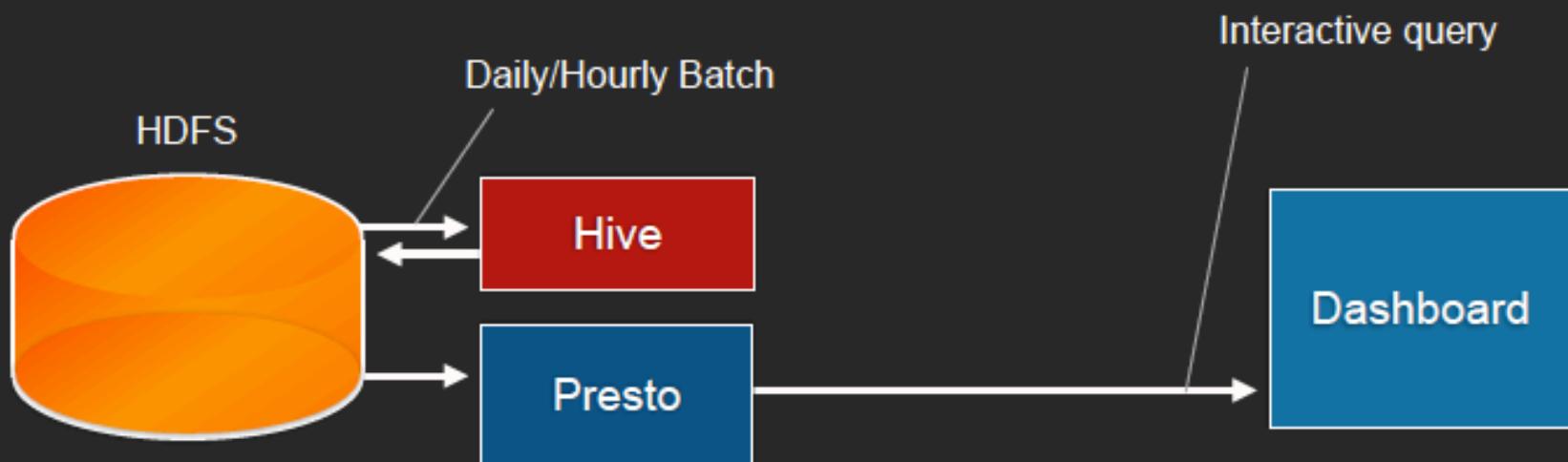
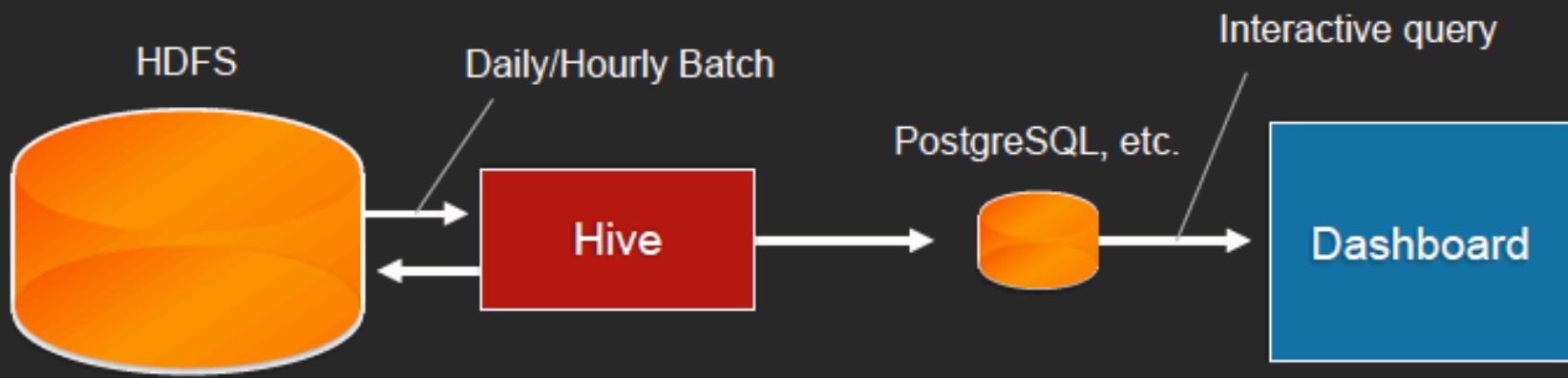
Động lực

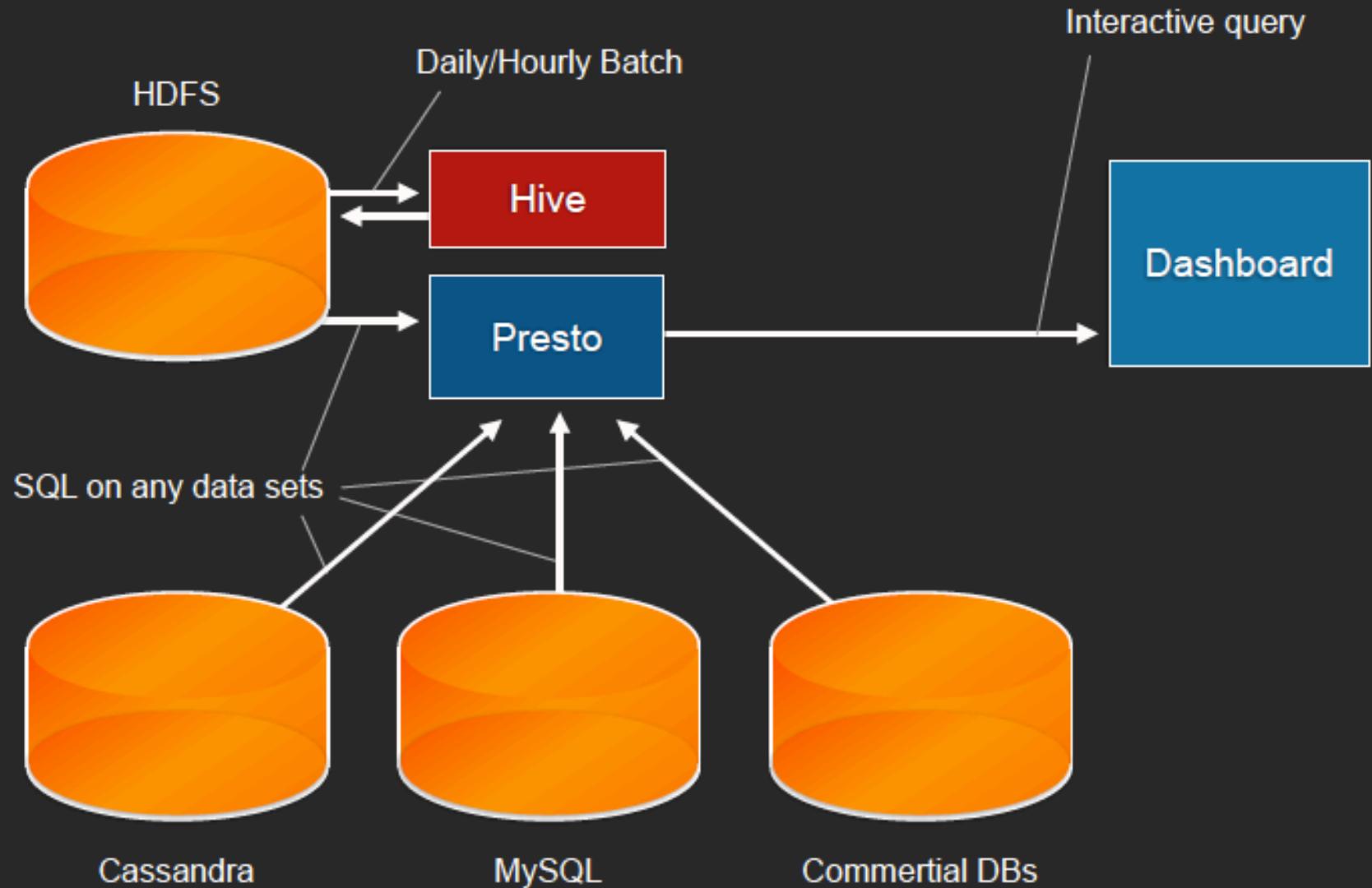
- Chúng tôi không thể trực quan hóa dữ liệu trong HDFS bằng bảng thông tin hoặc công cụ BI
 - vì Hive quá chậm (không tương tác)
 - hoặc kết nối ODBC không khả dụng/không ổn định
- Chúng tôi cần lưu trữ kết quả hàng ngày vào cơ sở dữ liệu tương tác để phản hồi nhanh (PostgreSQL, Redshift, v.v.)
 - Tính đến nay, chi phí DB tương tác ngày càng ít có khả năng mở rộng
- Một số dữ liệu không được lưu trữ trong HDFS
 - Chúng ta cần sao chép dữ liệu vào HDFS để phân tích

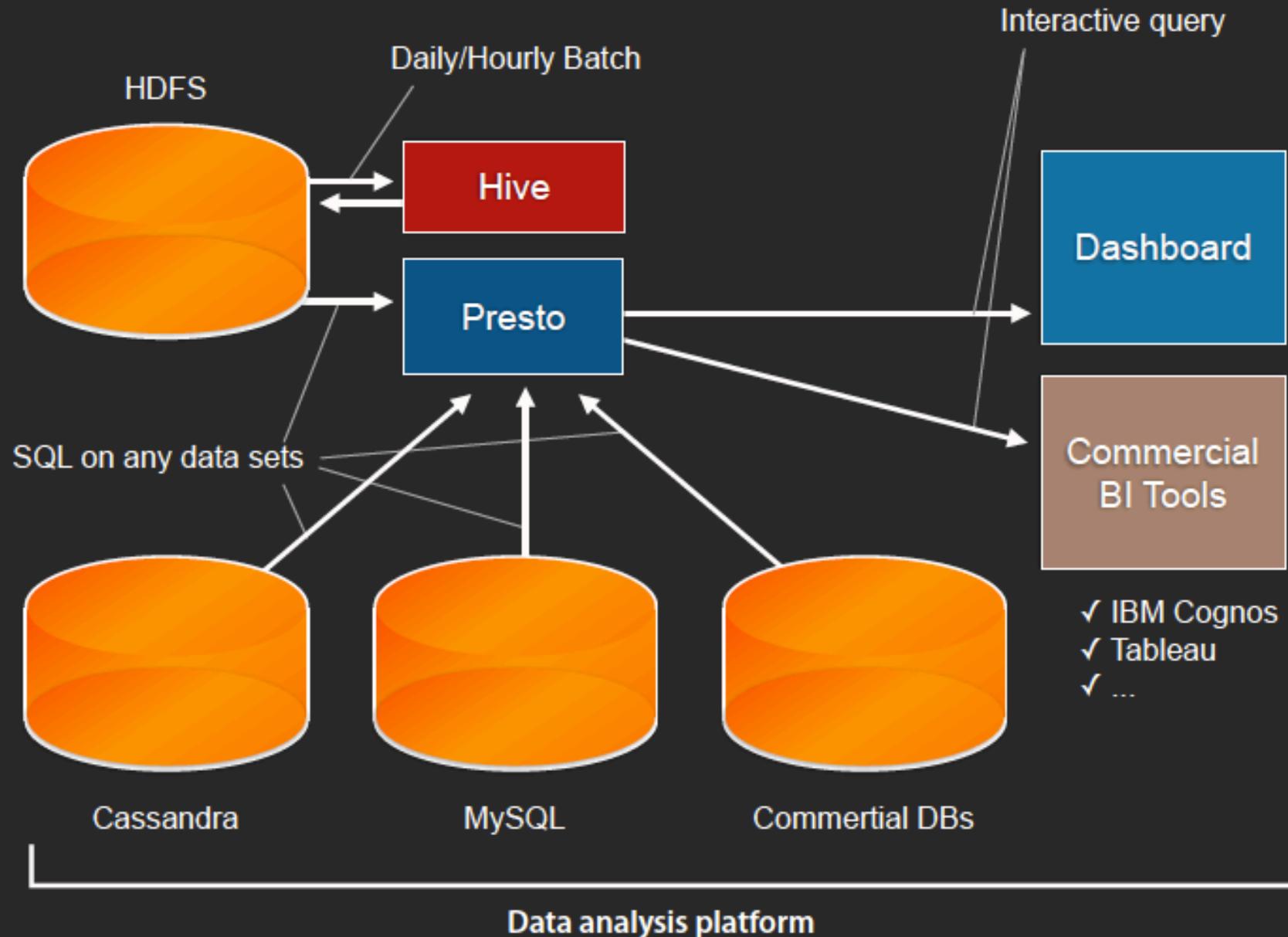
khả năng trích xuất nhanh chóng và dễ dàng những hiểu biết sâu sắc từ lượng lớn dữ liệu











Presto có thể làm gì?

- Công cụ truy vấn SQL phân tán mã nguồn mở đã được đưa vào sản xuất tại Facebook 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 hoặc bảng thông tin BI thương mại
 - Kết nối ODBC/JDBC đáng tin cậy
- Truy vấn trên nhiều nguồn dữ liệu như Hive, HBase, Cassandra hoặc thậm chí cả DB thương mại
 - Cơ chế bổ sung
- Tích hợp phân tích hàng loạt + trực quan hóa vào một nền tảng phân tích dữ liệu duy nhất

Triển khai nhanh chóng

- Facebook (2013)
 - Nhiều vùng địa lý
 - Mở rộng tới 1.000 nút
 - Được sử dụng tích cực bởi hơn 1.000 nhân viên chạy hơn 30.000 truy vấn mỗi ngày
 - Xử lý 1PB/ngày

NETFLIX

facebook

LinkedIn



TERADATA®



Amazon Athena

zuora

YAHOO!
JAPAN



FreeWheel

amazon

**mercado
Libre.com**

LinkedIn

jampp

**Marin
SOFTWARE**

looker

comcast

wix

**TREASURE
DATA**

airbnb

WB

Walmart

Alibaba Group

**U
BER**

slack

Bloomberg

Groupon

GREE

**cogo
labs**

FINRA

twitter

trulia

shopify

Atlassian

AdRoll

Pinterest

openspan

shazam

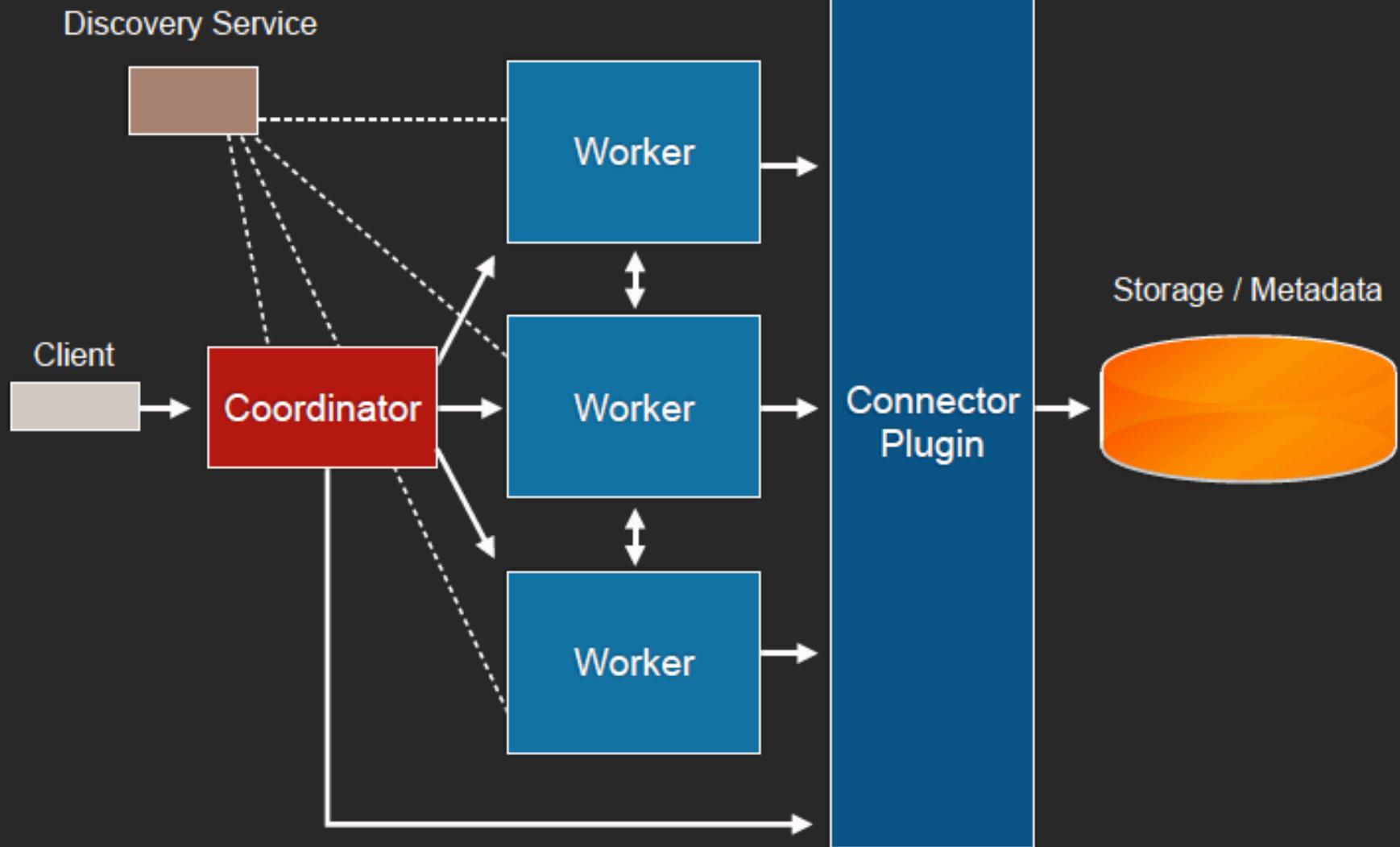
CUEBIQ

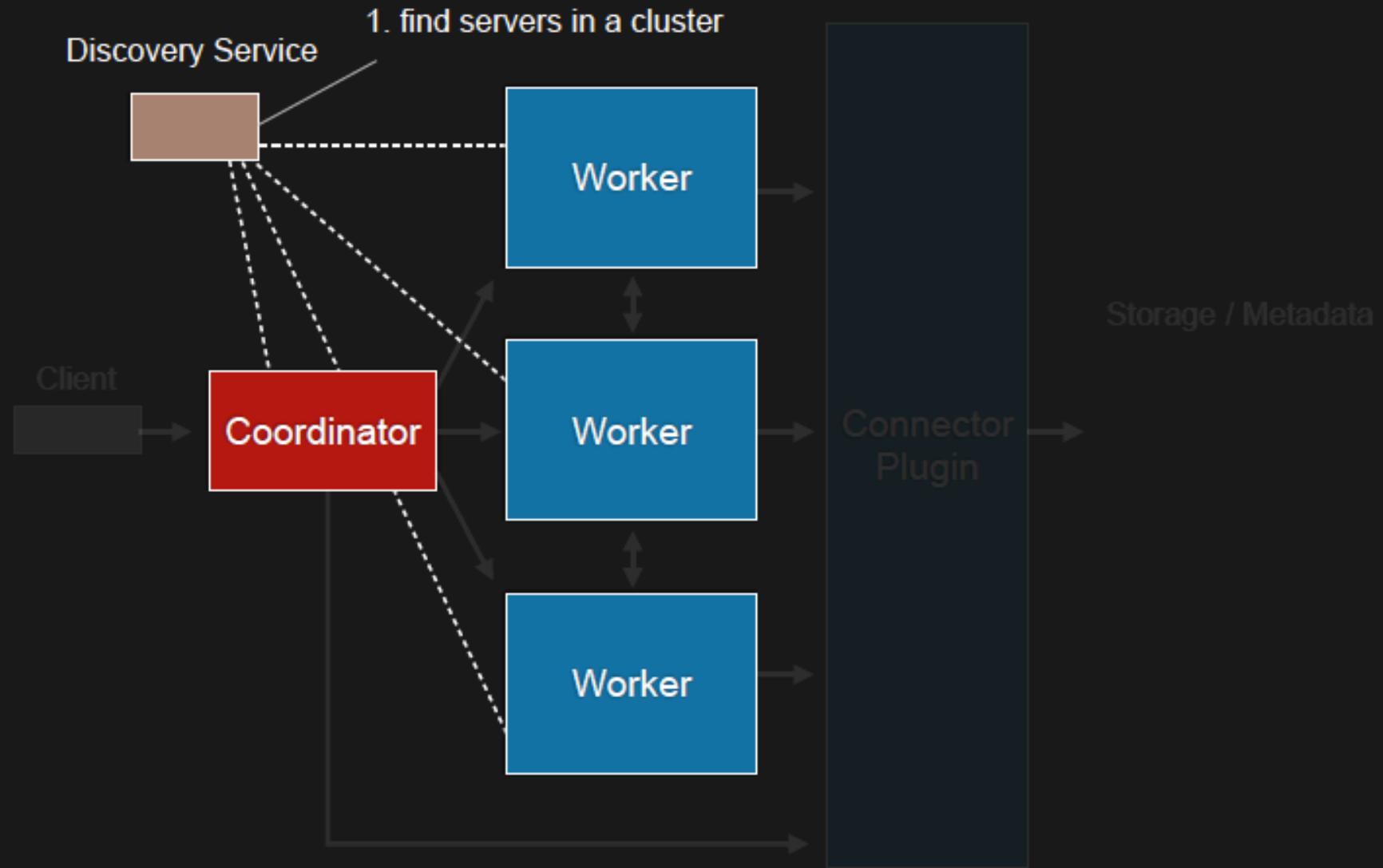
NASDAQ

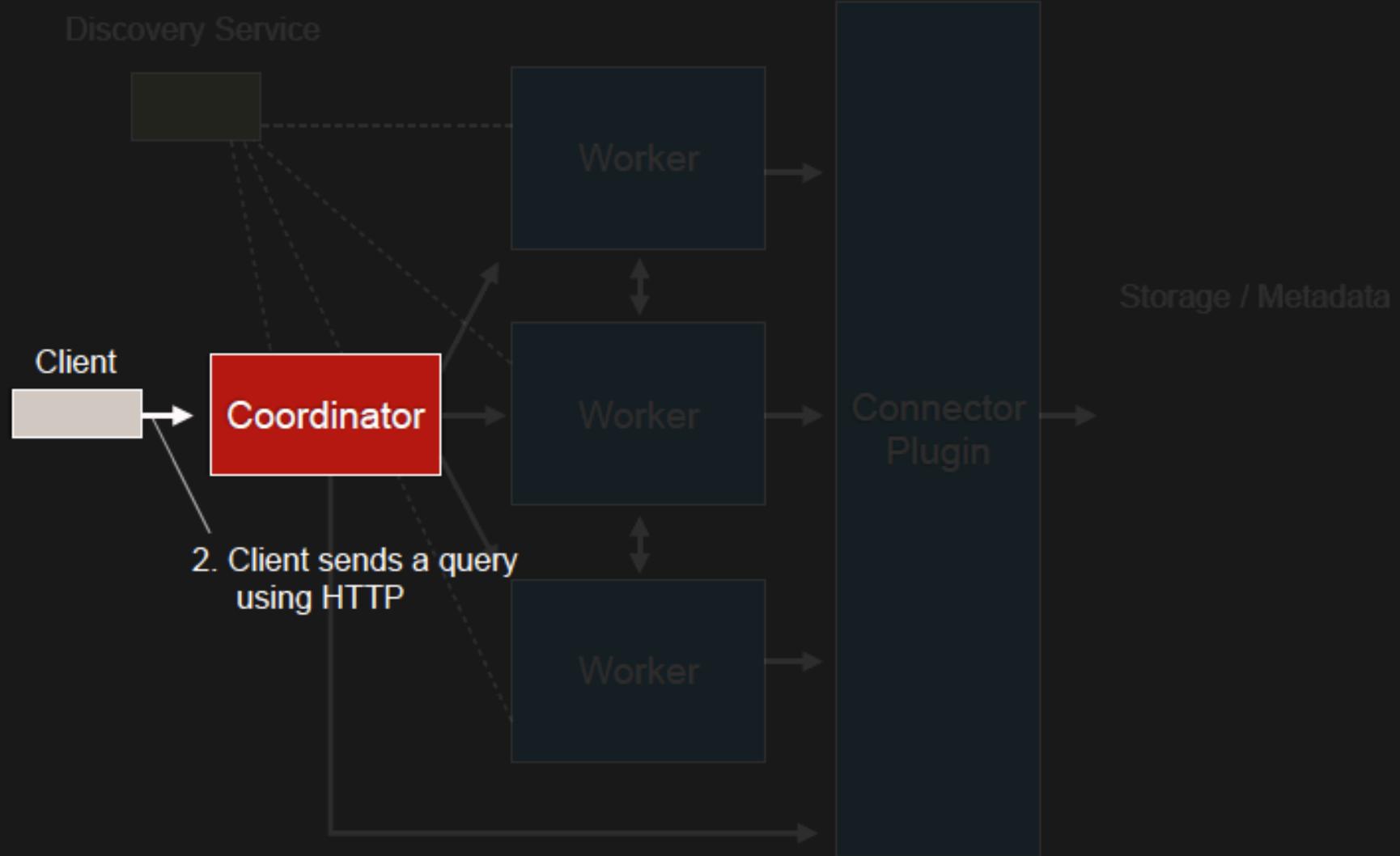
Dropbox

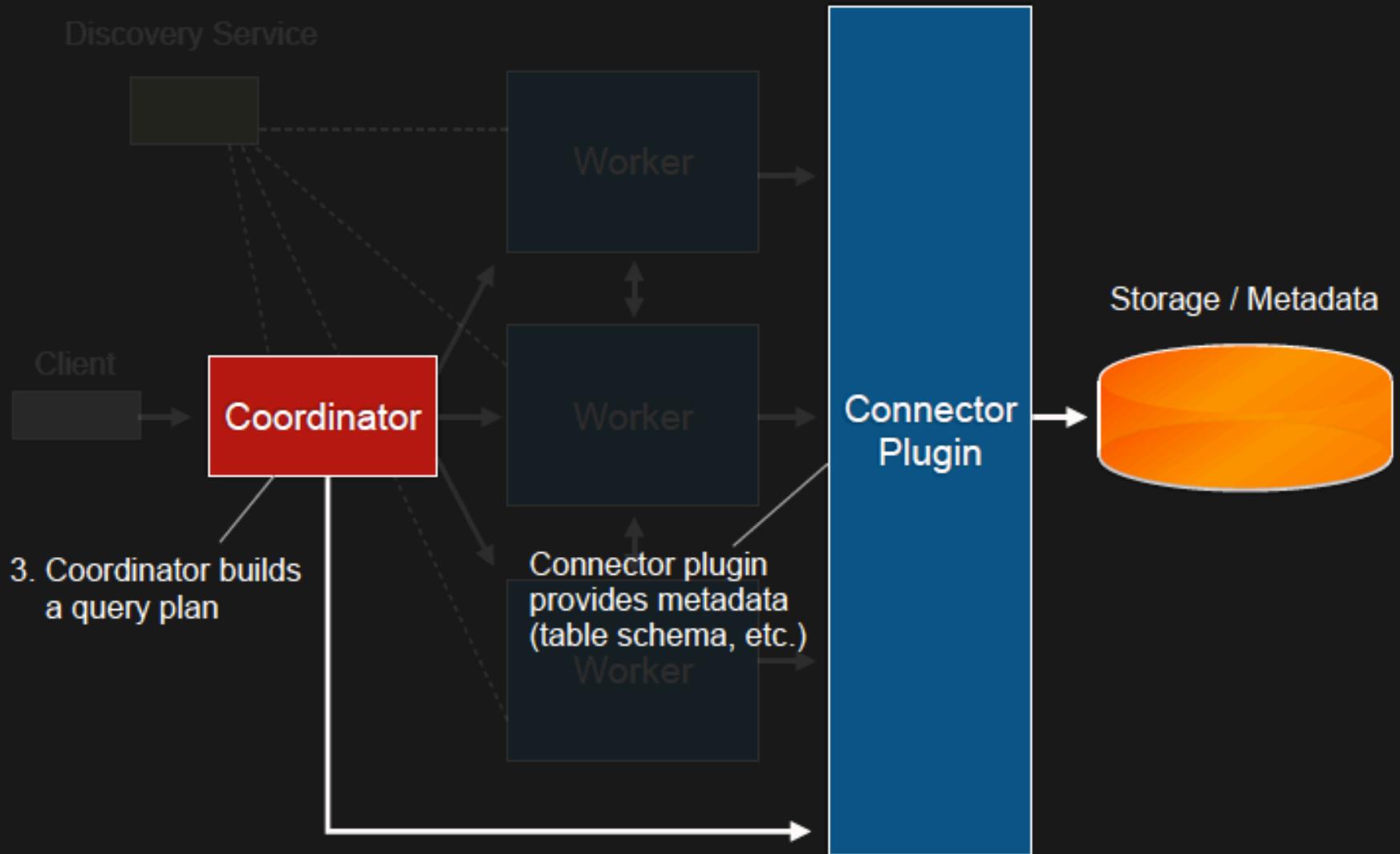
NETFLIX

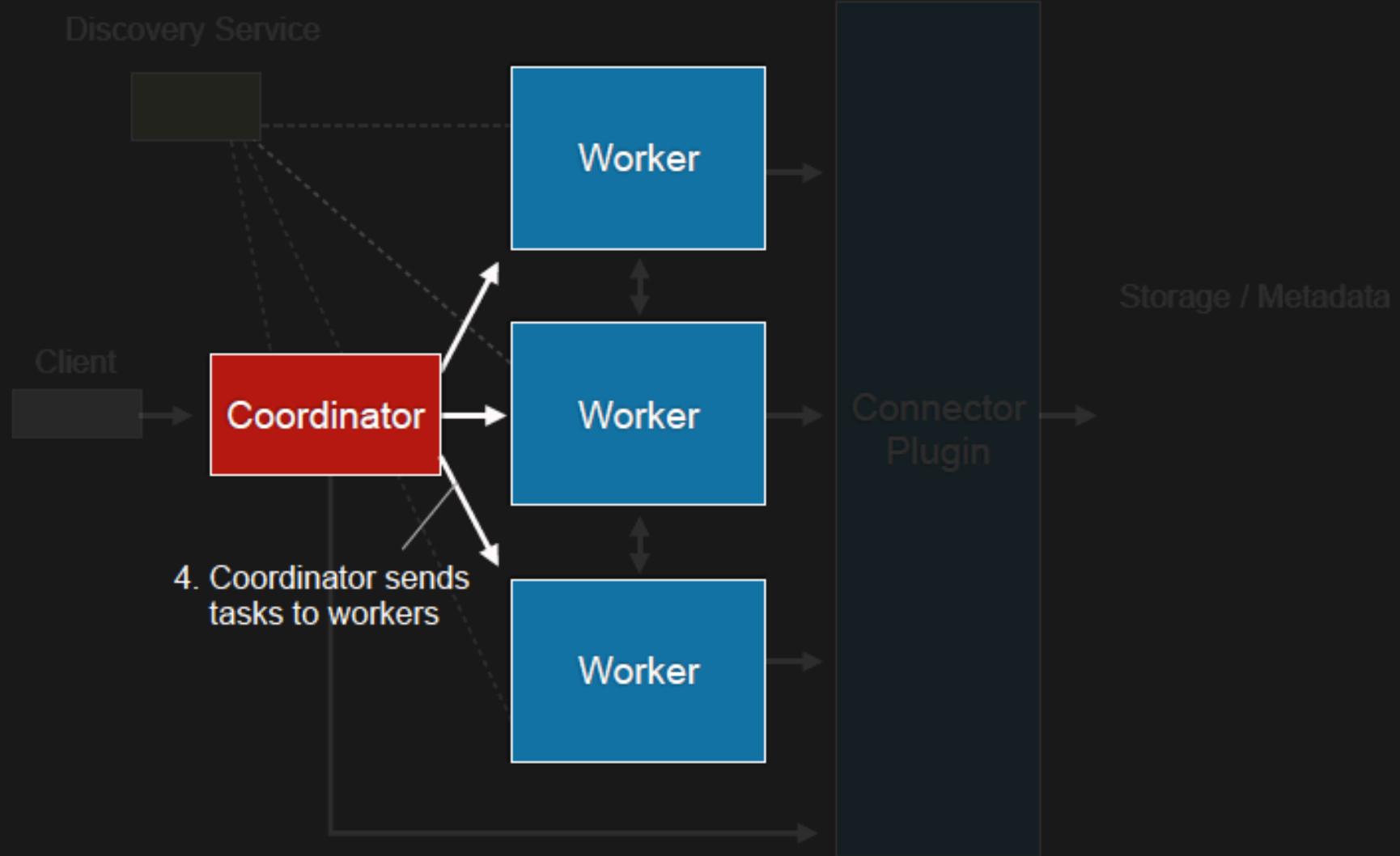
Kiến trúc Presto

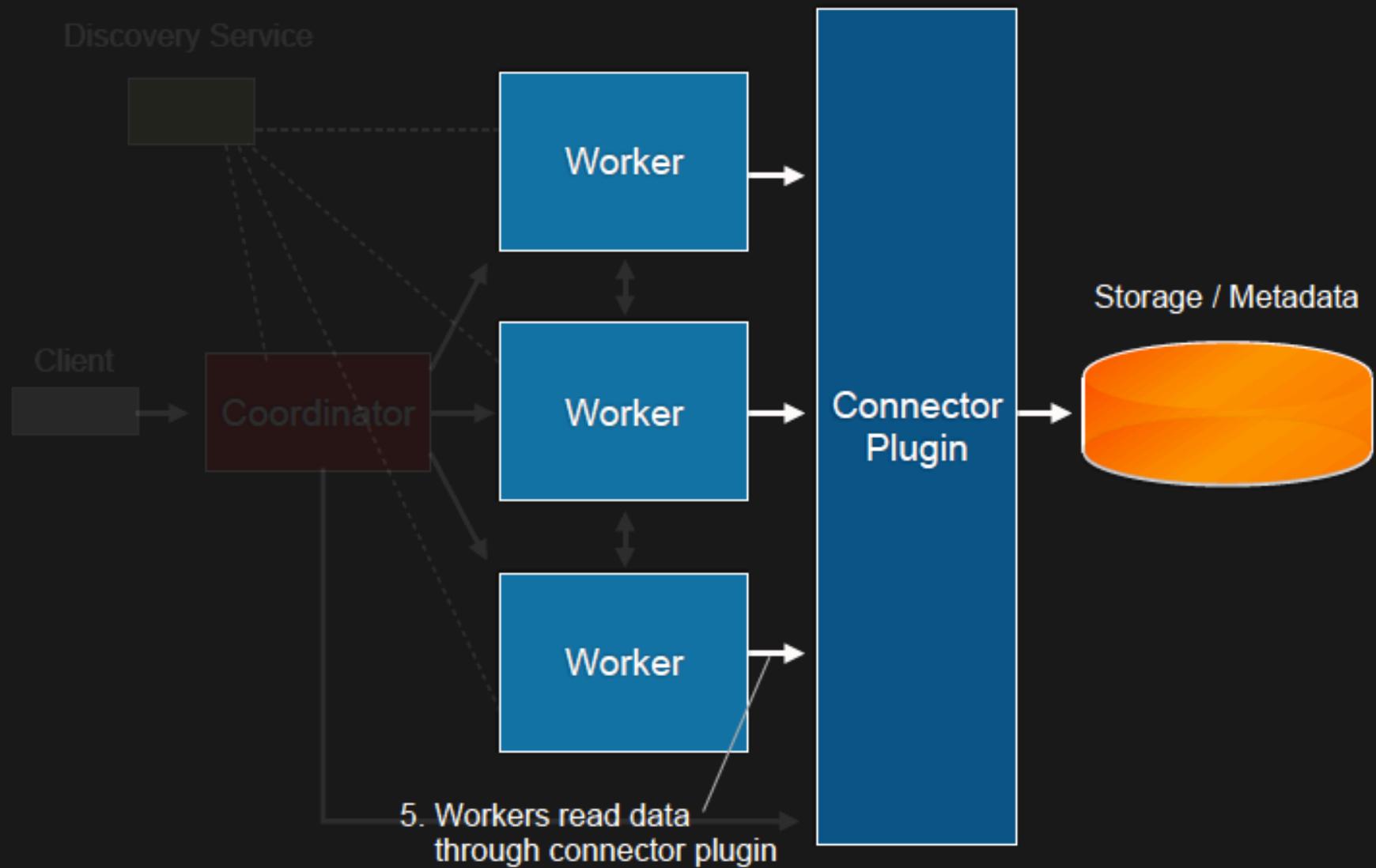


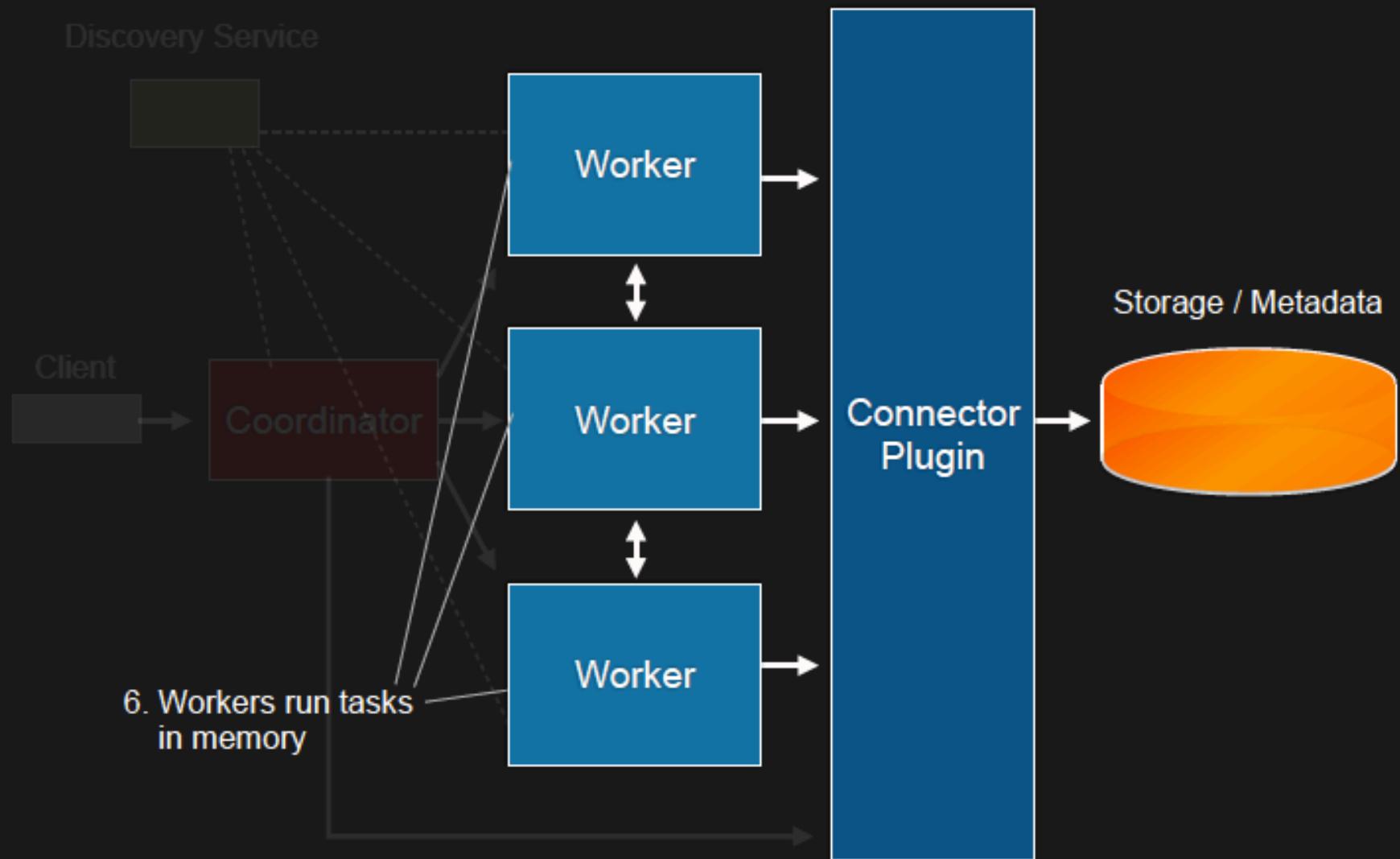


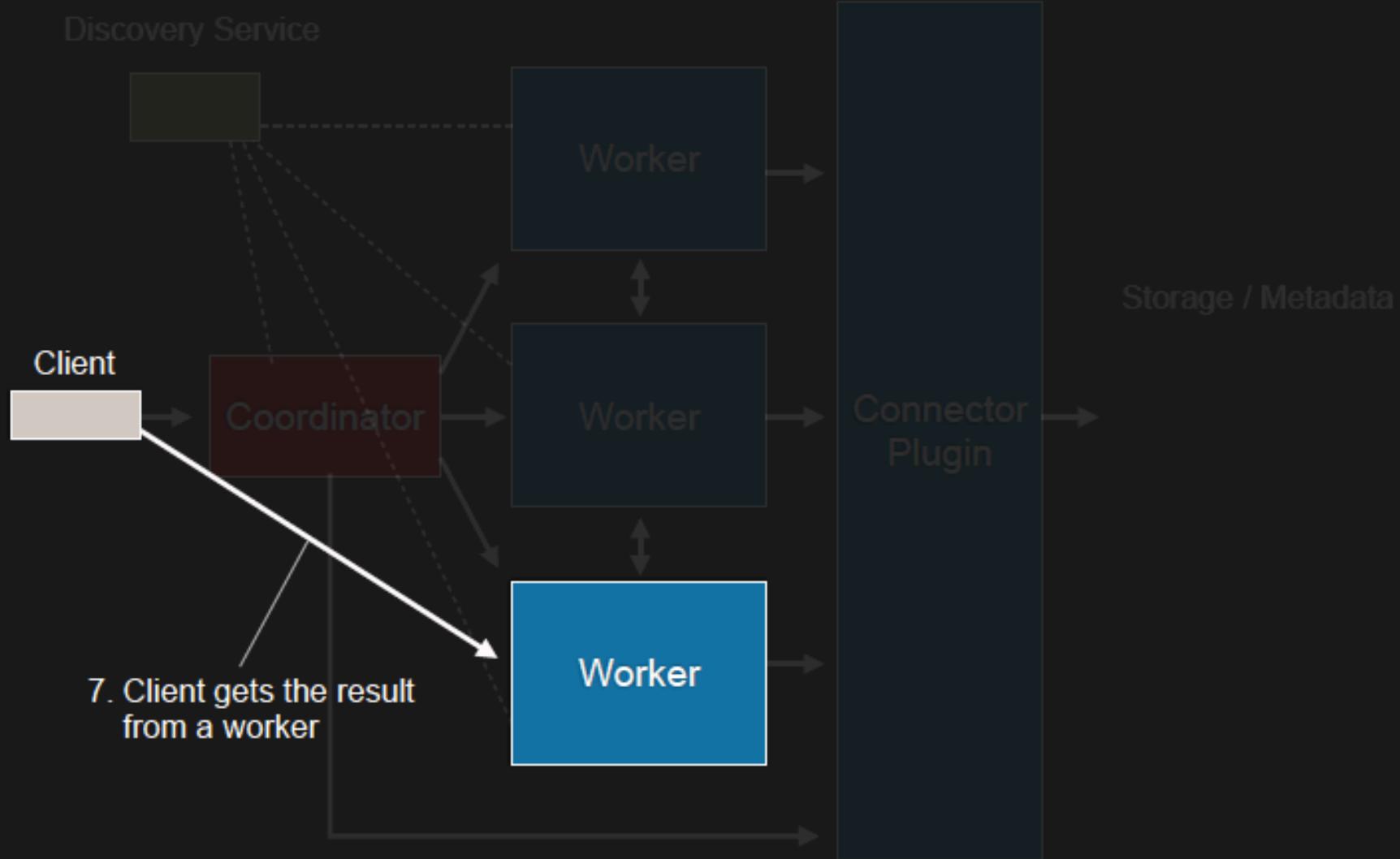


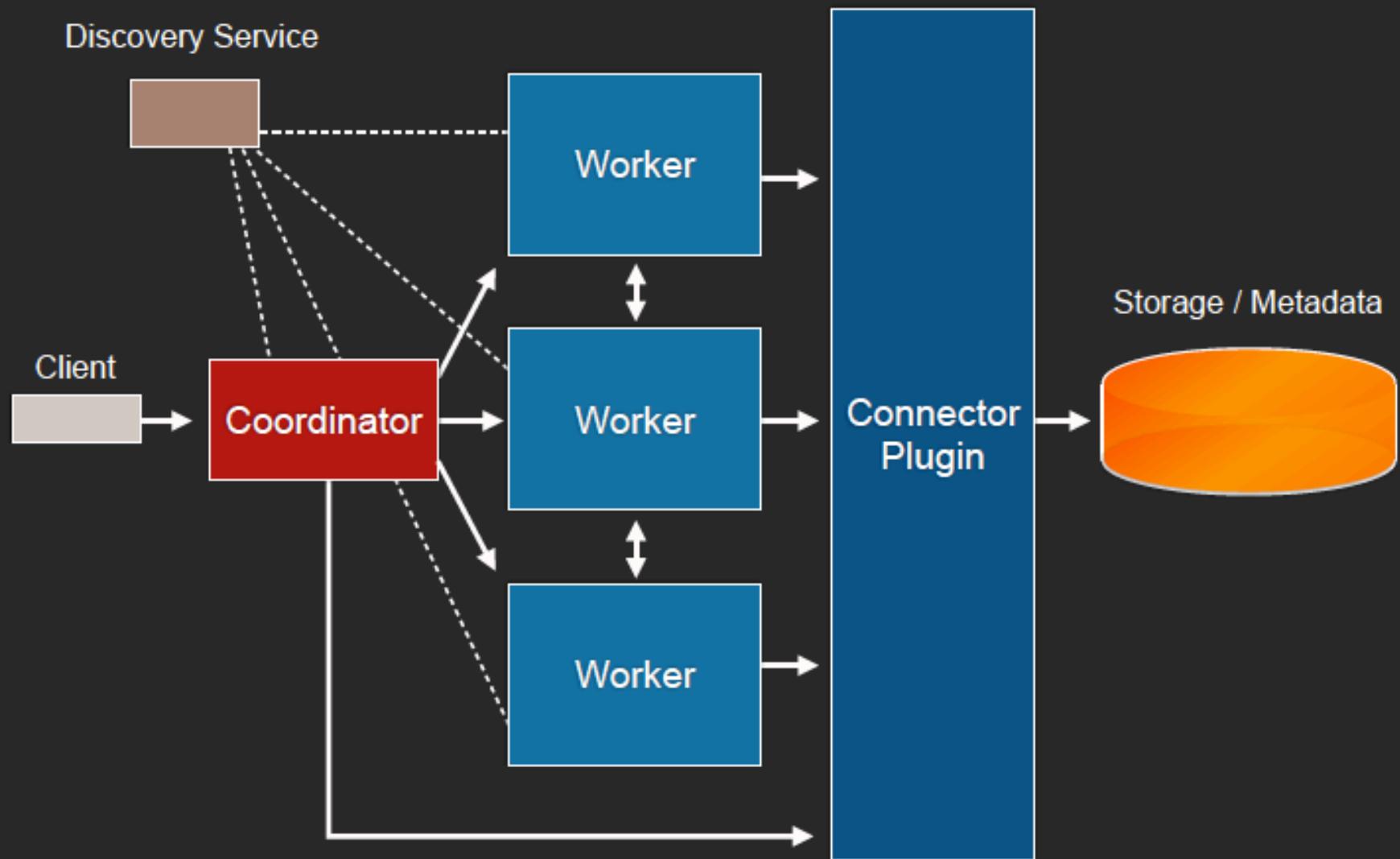








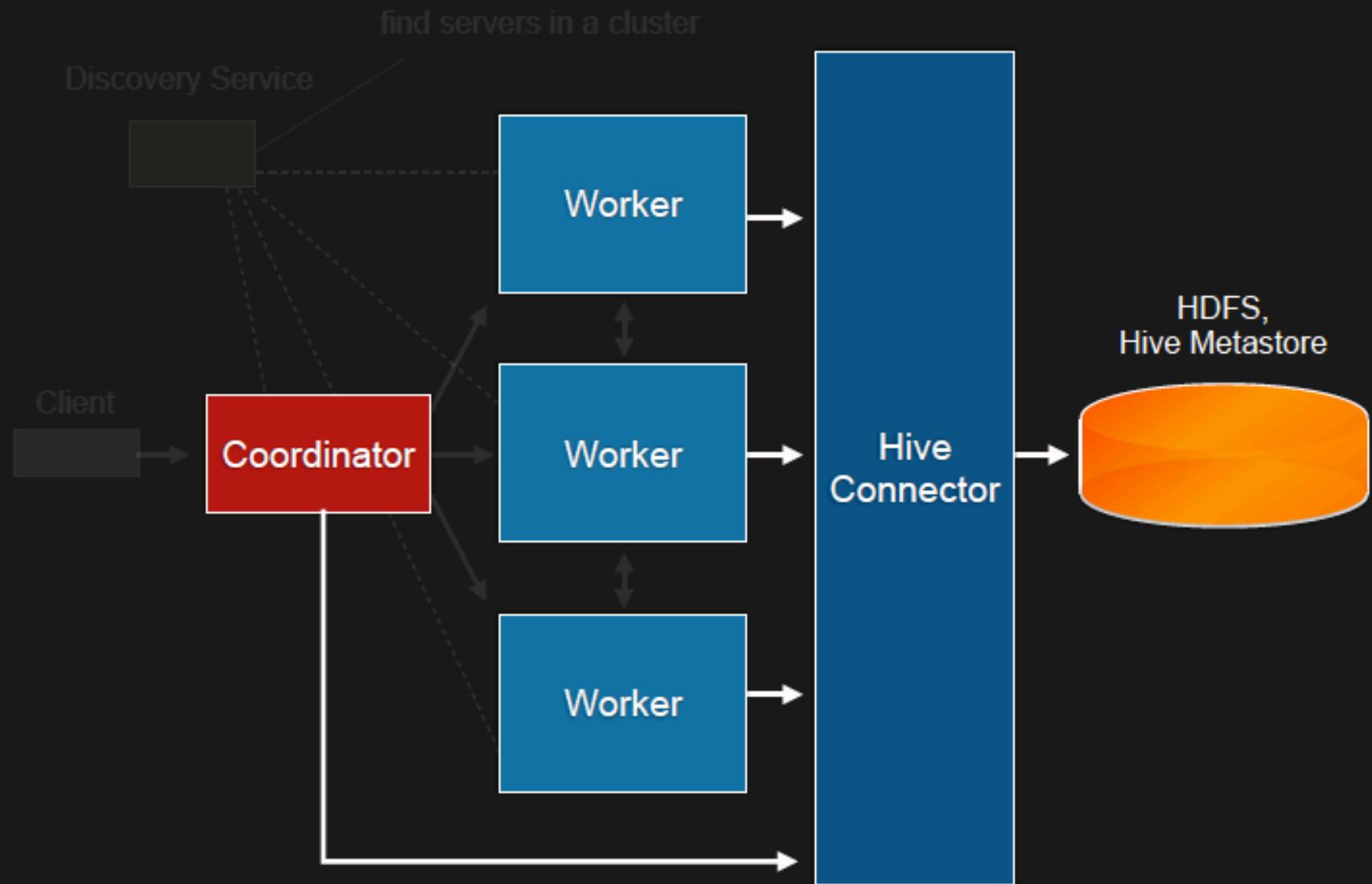




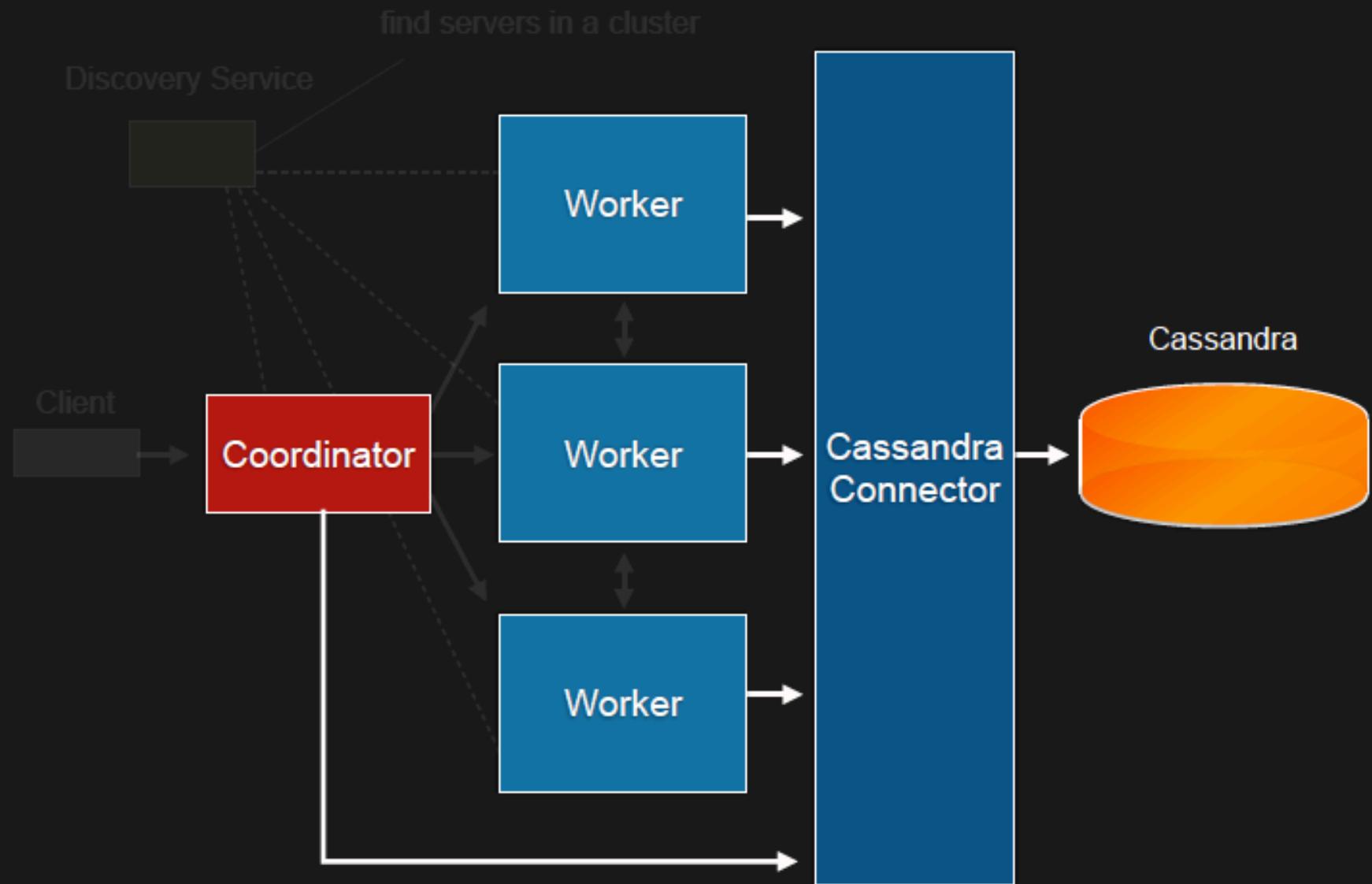
Đầu nối Presto

- Trình kết nối là phần bổ sung cho Presto
 - viết bằng Java
- Truy cập vào bộ lưu trữ và siêu dữ liệu
 - cung cấp lược đồ bảng cho điều phối viên
 - cung cấp các hàng trong bảng cho công nhân
- Triển khai
 - Đầu nối tổ ong
 - Đầu nối Cassandra
 - MySQL thông qua trình kết nối JDBC (bản phát hành trước)
 - Hoặc trình kết nối của riêng bạn

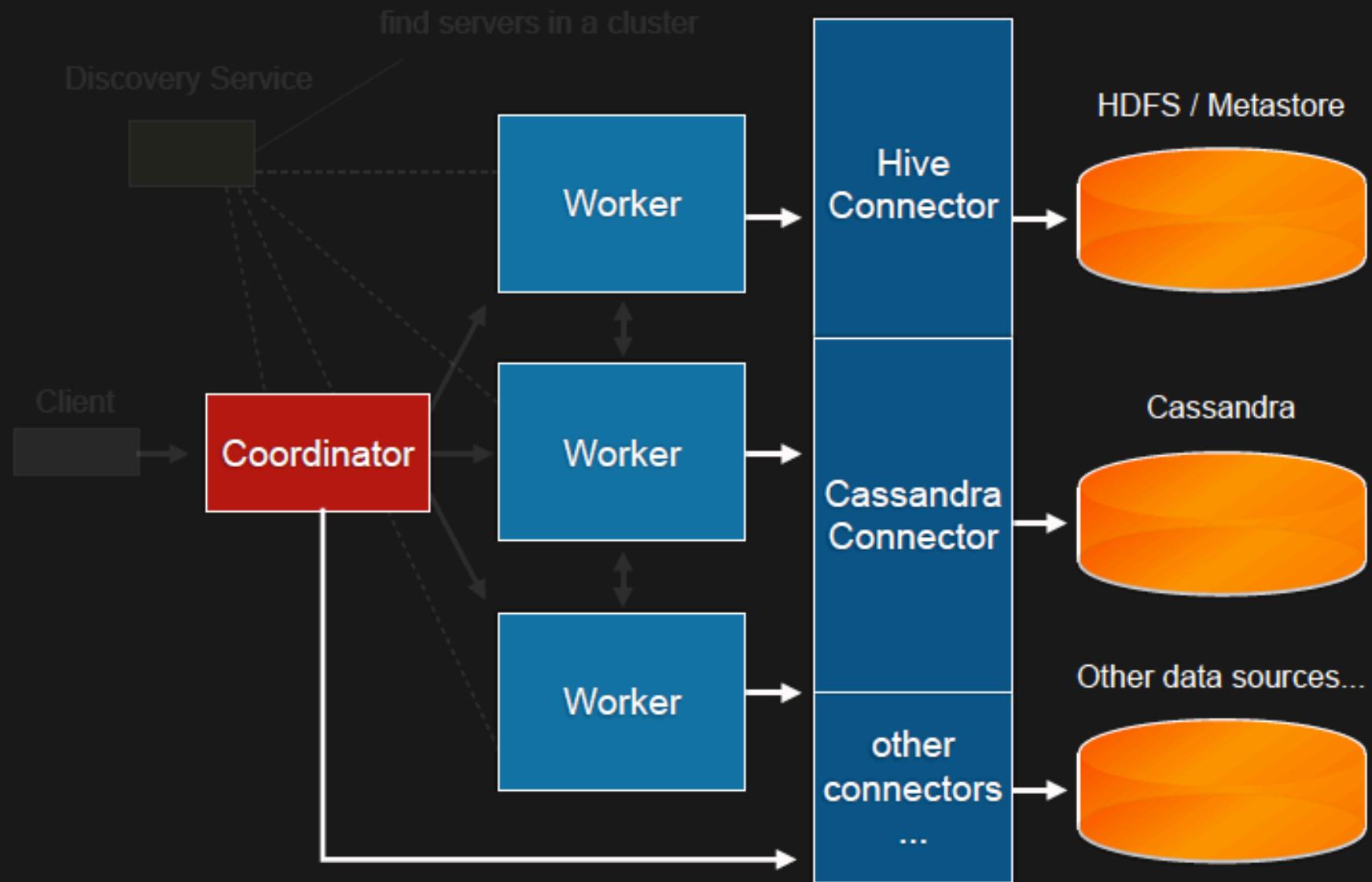
Hive connector



Cassandra connector



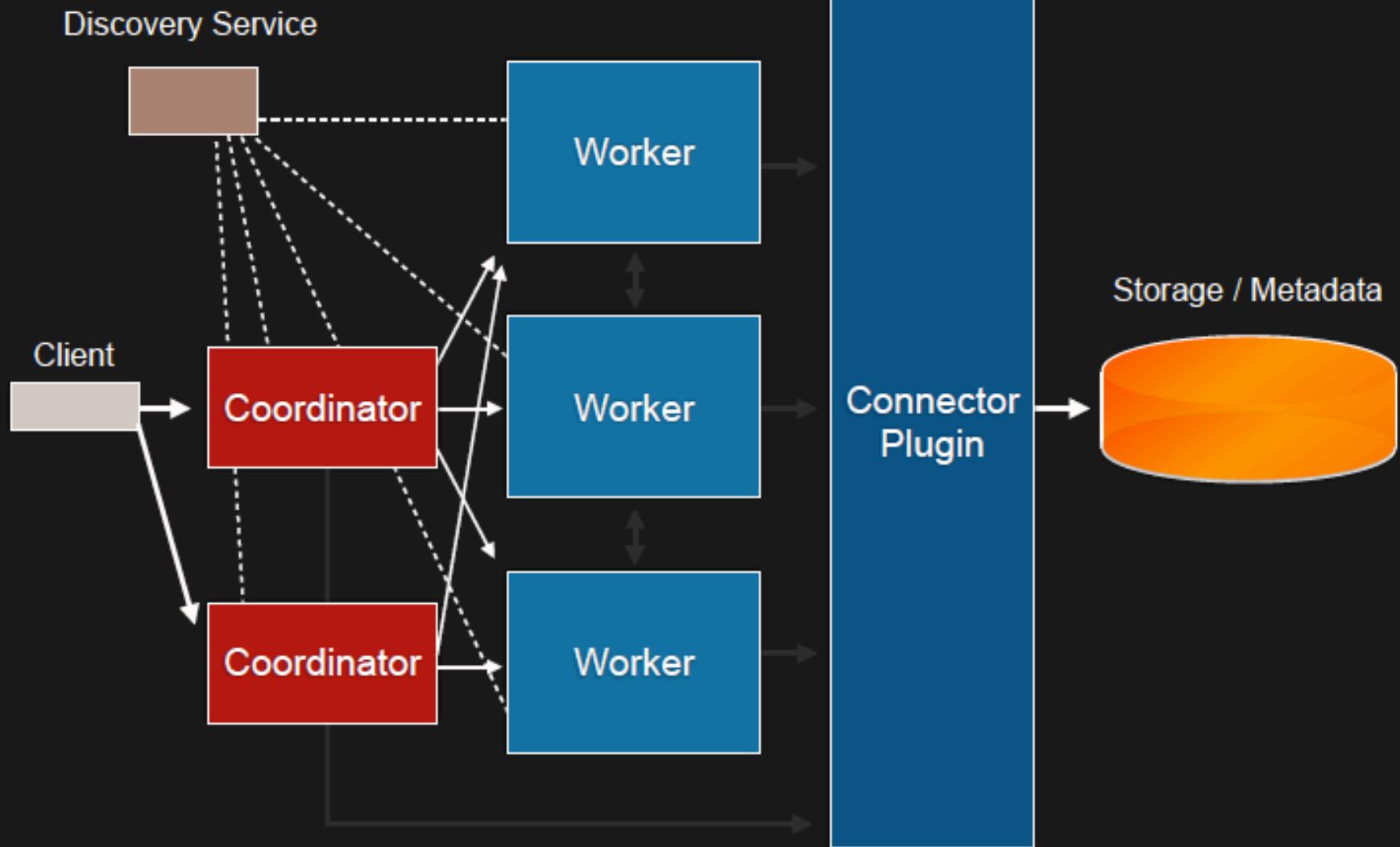
Multiple connectors in a query



Kiến trúc phân tán

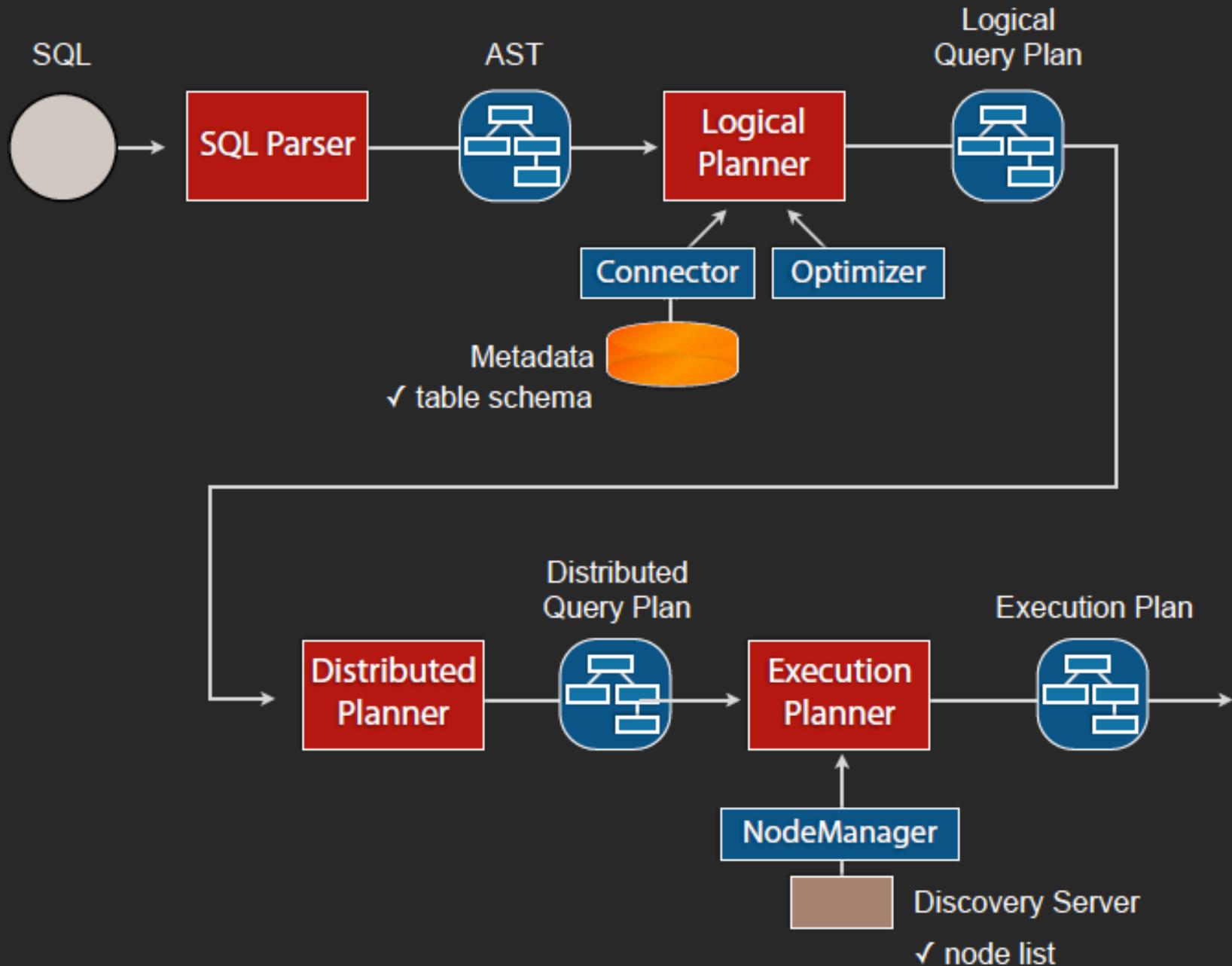
- 3 loại máy chủ:
 - Điều phối viên, nhân viên, dịch vụ khám phá
- Nhận dữ liệu/siêu dữ liệu thông qua các plugin kết nối.
 - Presto KHÔNG phải là cơ sở dữ liệu
 - Presto cung cấp SQL cho các kho dữ liệu hiện có
- Giao thức máy khách là HTTP + JSON
 - Liên kết ngôn ngữ: Ruby, Python, PHP, Java (JDBC),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

Coordinator HA



Mô hình thực thi Presto

- Presto KHÔNG phải là MapReduce
- Kế hoạch truy vấn của Presto dựa trên DAG
 - giống Apache Tez hoặc cơ sở dữ liệu MPP truyền thống hơn
- Truy vấn chạy như thế nào?
 - Điều phối viên
 - Trình phân tích cú pháp SQL
 - Công cụ lập kế hoạch truy vấn
 - Người lập kế hoạch thực hiện
 - Công nhân
 - Lập lịch thực hiện nhiệm vụ



Query Planner

SQL

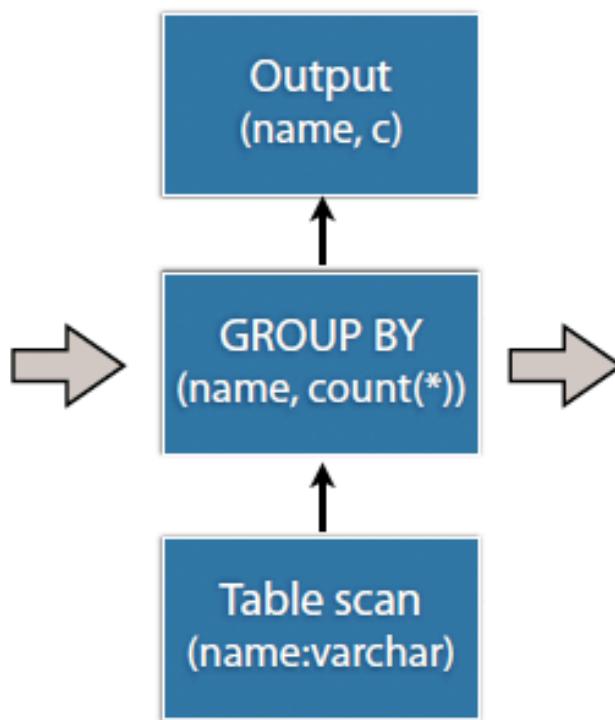
```
SELECT  
    name,  
    count(*) AS c  
FROM impressions  
GROUP BY name
```

+

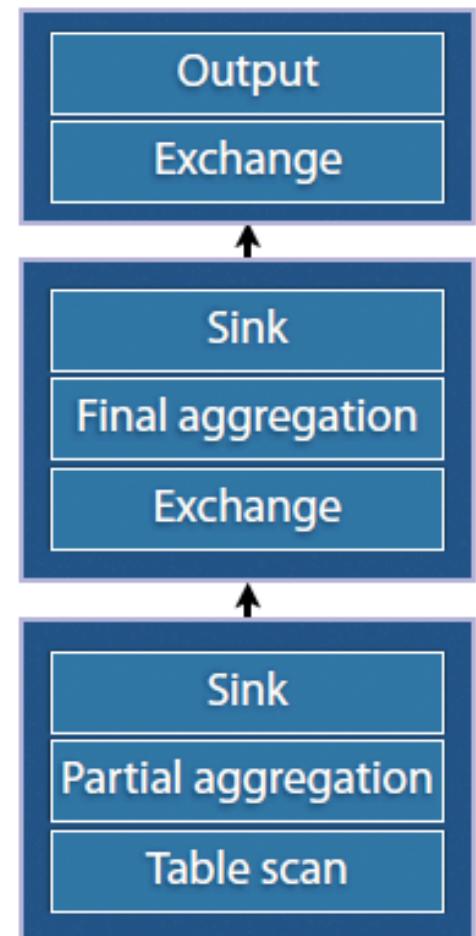
Table schema

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

Logical query plan



Distributed query plan



Query Planner - Stages

Stage-0

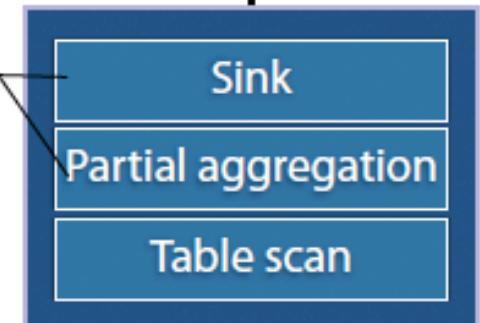
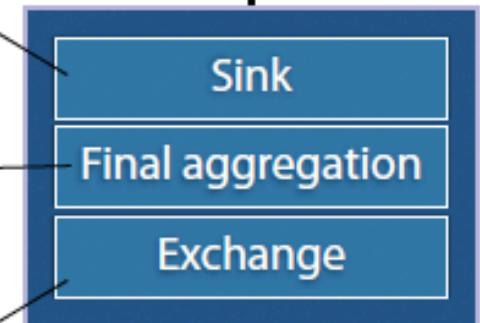
**inter-worker
data transfer**

Stage-1

**pipelined
aggregation**

Stage-2

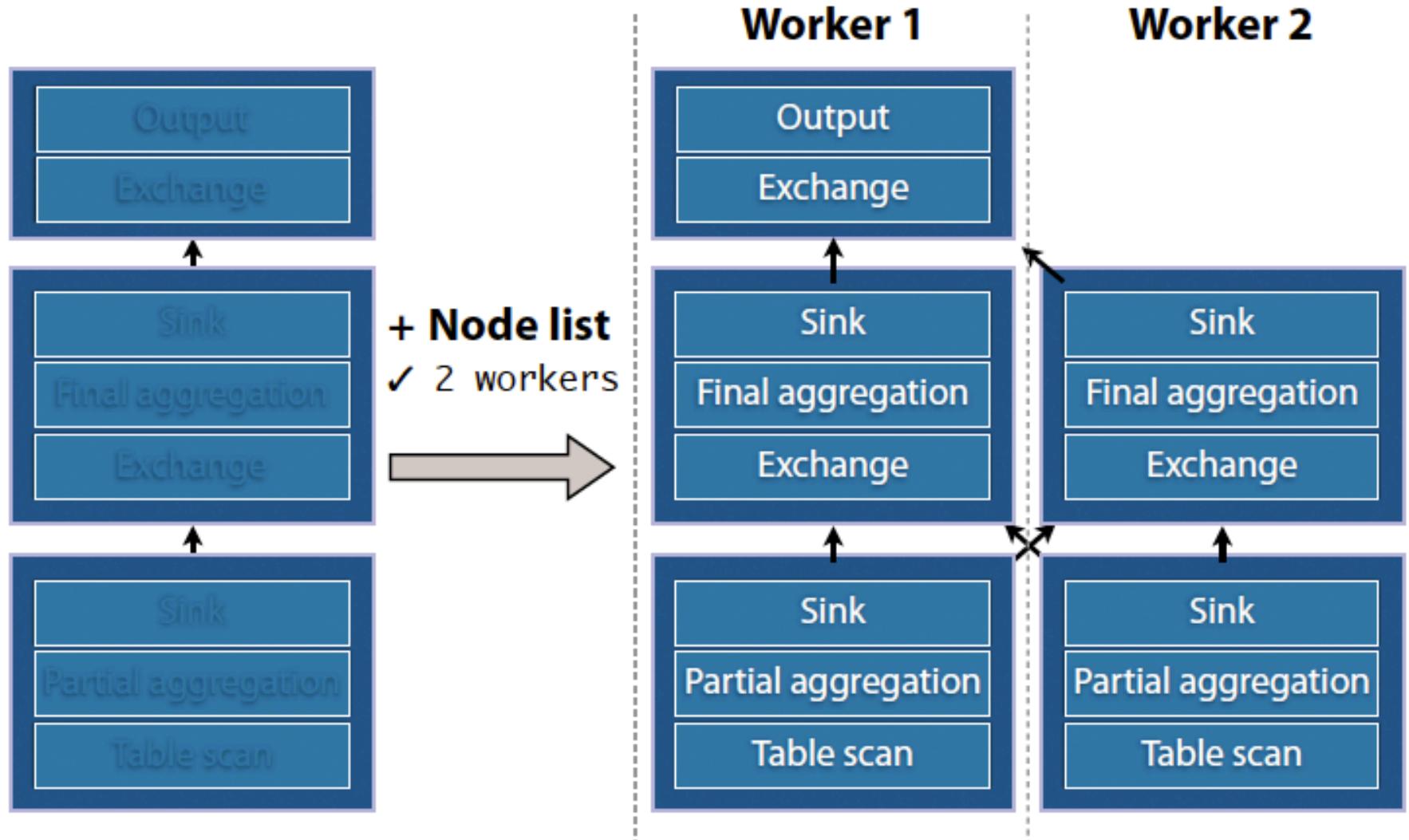
**inter-worker
data transfer**



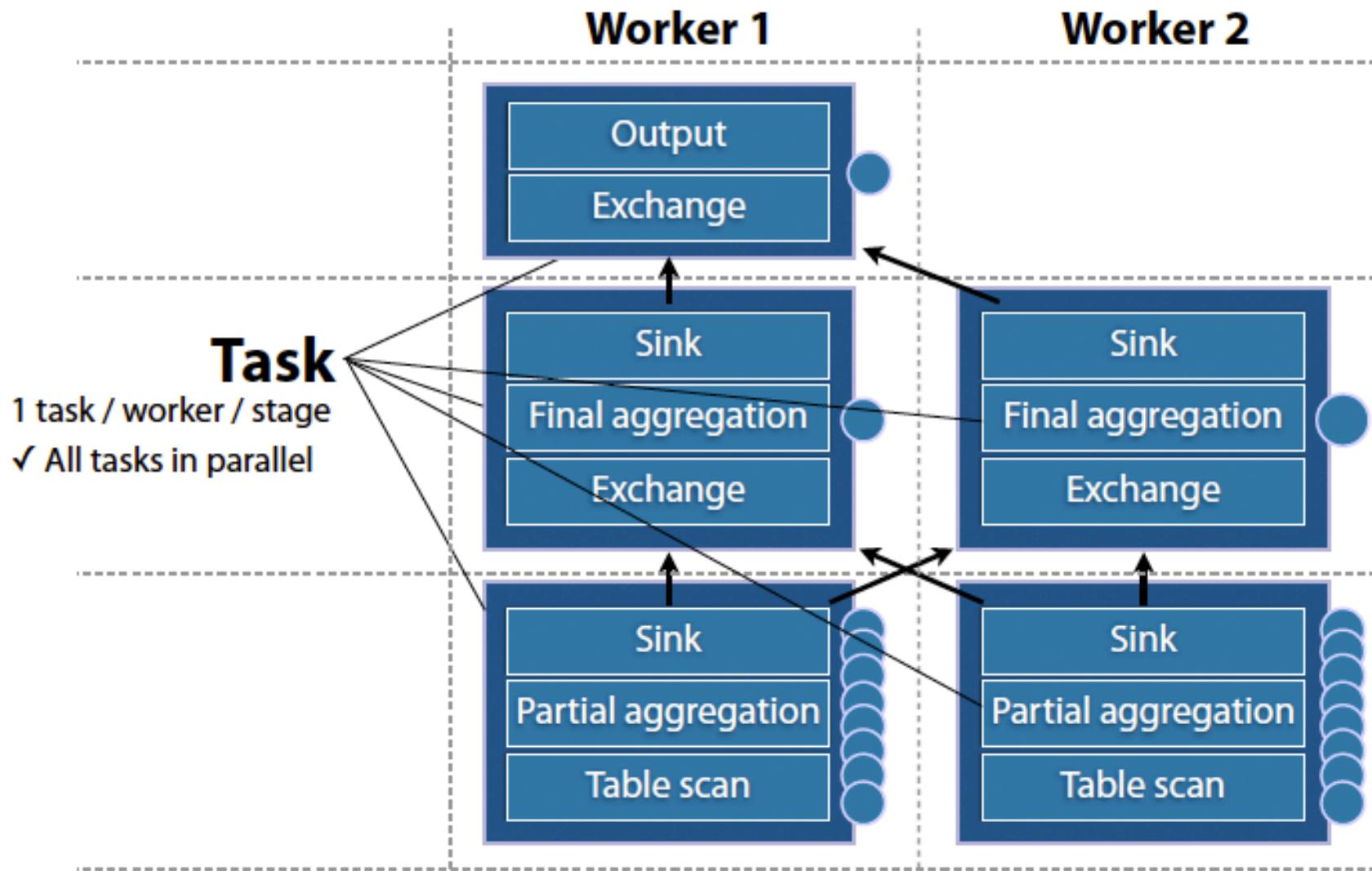
Giai đoạn

- 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 bộ dữ liệu đầu vào khác nhau
 - Truyền dữ liệu trong bộ nhớ đệm (xáo trộn) giữa các giai đoạn để cho phép trao đổi dữ liệu
- Việc xáo trộn sẽ tăng thêm độ trễ, sử dụng hết bộ nhớ đệm và tiêu tốn nhiều CPU

Execution Planner

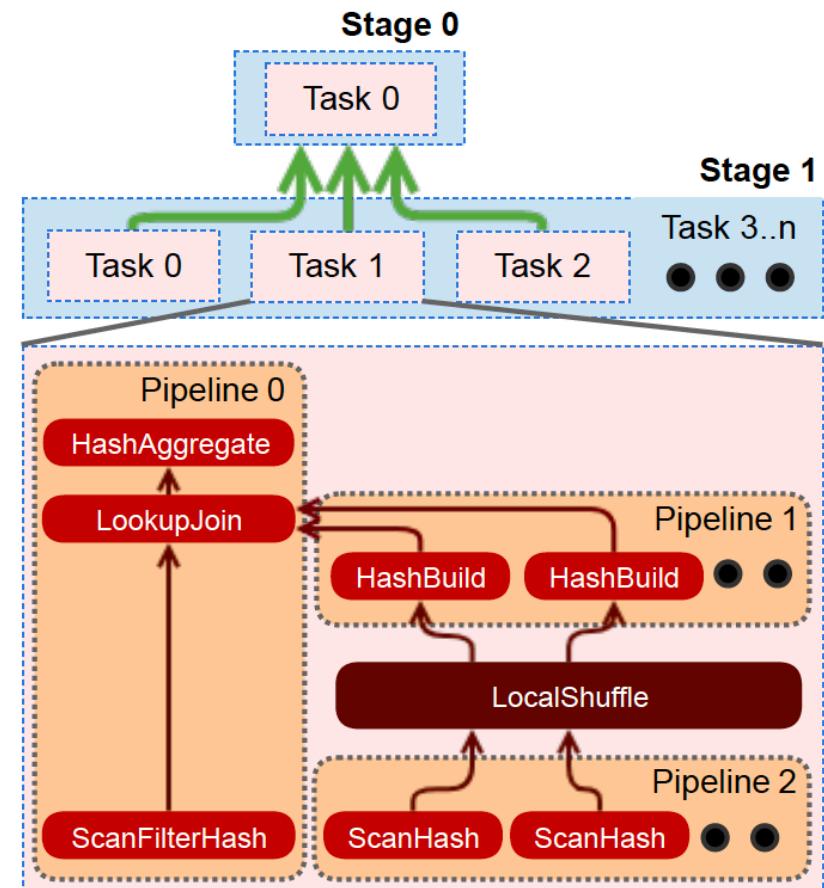


Execution Planner - Tasks



Nhiệm vụ

- Điều phối viên phân phối các giai đoạn kế hoạch cho người lao động dưới dạng các nhiệm vụ thực thi
- 1 công việc/ 1 công nhân/ 1 công đoạn
- Một tác vụ có thể có nhiều đường dẫn
- Một đường ống bao gồm một chuỗi các nhà khai thác



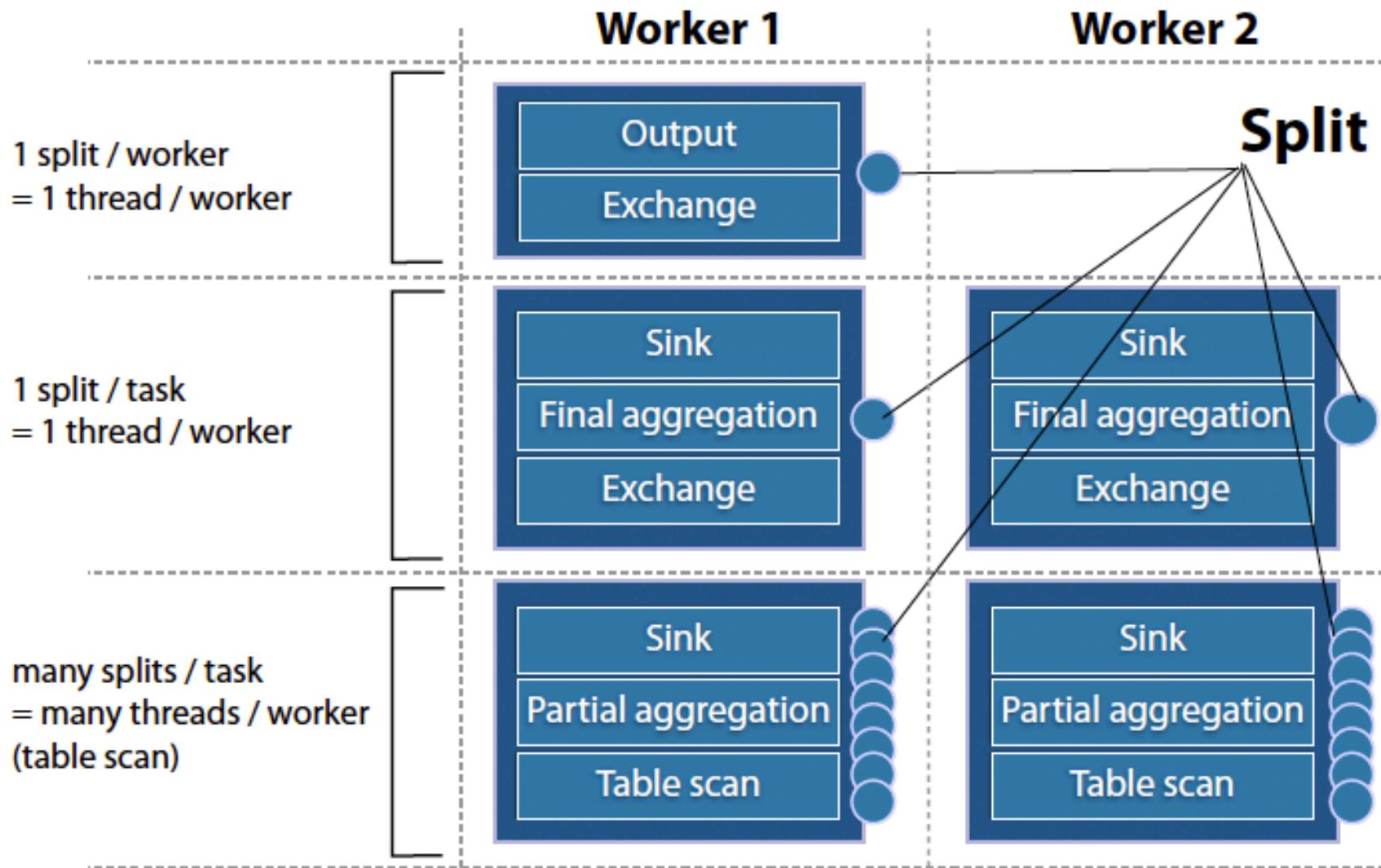
Lên lịch

- Để thực hiện một truy vấn, Bộ lập lịch đưa ra hai bộ quyết định lập lịch
- Lên lịch giai đoạn
 - tất cả cùng một lúc: giảm thiểu thời gian đồng hồ treo tường bằng cách lên lịch đồng thời cho tất cả các giai đoạn thực hiện
 - dữ liệu được xử lý ngay khi có sẵn
 - mang lại lợi ích cho các trường hợp sử dụng nhạy cảm với độ trễ như Phân tích tương tác, Phân tích nhà phát triển/nhà quảng cáo và Thủ nghiệm A/B
 - Theo giai đoạn:
 - Sau khi một giai đoạn được lên lịch theo chính sách, nó sẽ bắt đầu phân công 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

Lập kế hoạch nhiệm vụ

- Giai đoạn được chia thành giai đoạn lá và giai đoạn trung gian
- **Giai đoạn lá**
 - bộ lập lịch tác vụ có tính đến các ràng buộc do mạng và trình kết nối áp đặt khi phân công nhiệm vụ cho các nút công nhân
 - các nút lưu trữ và công nhân cùng vị trí
 - Các ràng buộc về bố cục dữ liệu của trình kết nối
- **Giai đoạn trung gian**
 - Nhiệm vụ cho các giai đoạn trung gian có thể được đặt trên bất kỳ nút công nhân nào
 - Engine vẫn cần quyết định số lượng nhiệm vụ cần được lên lịch cho mỗi giai đoạn

Execution Planner - Split



Chia lịch trình

- Phần tách là các phần xử lý không rõ ràng đối với một đoạn dữ liệu có thể định địa chỉ
 - trong hệ thống lưu trữ bên ngoài
 - hoặc kết quả trung gian do người lao động khác tạo ra
 - Ví dụ. Đọc từ HDFS
 - Phần tách là đường dẫn tệp và phần bù cho một vùng của tệp
- Chia bài tập
 - Sự phân chia được phân công cho từng nhiệm vụ một cách lười biếng
 - Các phần tách được liệt kê khi truy vấn thực thi, không được liệt kê ở phía trước
 - Các truy vấn có thể bắt đầu tạo ra kết quả mà không cần xử lý tất cả dữ liệu
 - Việc phân chia được chỉ định cho công nhân có hàng đợi ngắn nhất
 - Giảm mức sử dụng bộ nhớ siêu dữ liệu trên điều phối viên

Tối ưu hóa truy vấn: Bố cục dữ liệu

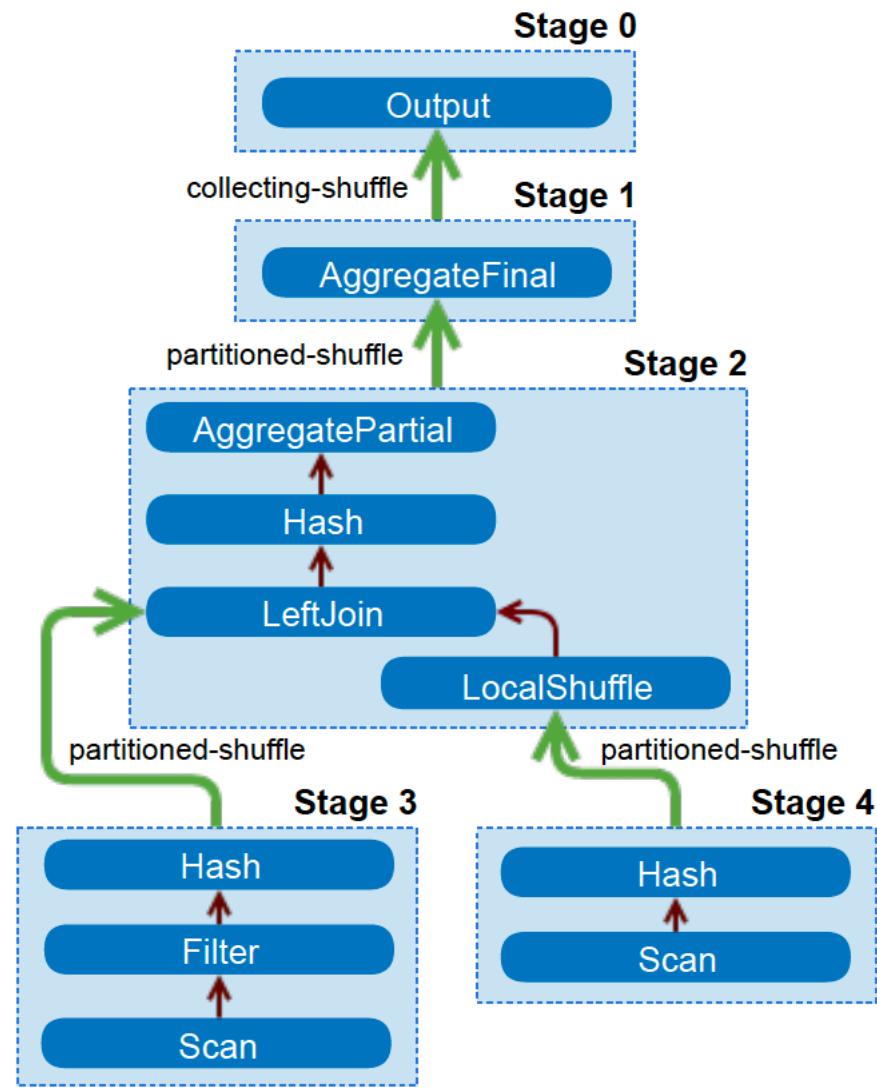
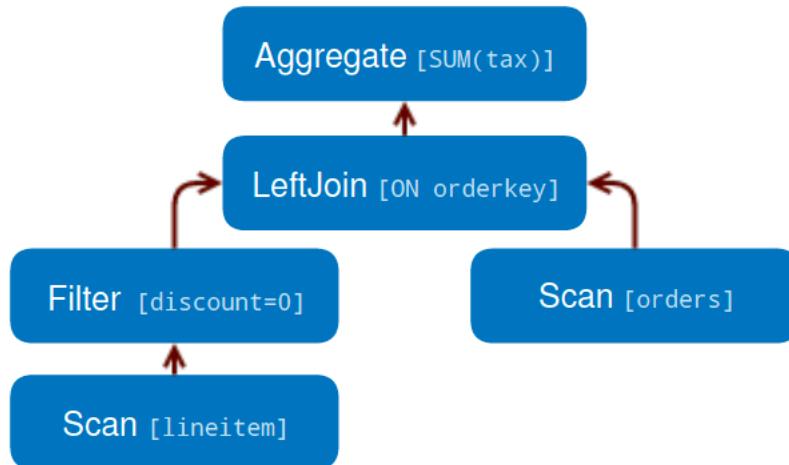
- Trình tối ưu hóa tận dụng bố cục vật lý của dữ liệu
 - Thuộc tính: phân vùng, sắp xếp, nhóm, chỉ mục
- Bảng có thể có nhiều bố cục với các thuộc tính khác nhau
 - Bố cục có thể có một tập hợp con các cột hoặc dữ liệu
- Trình tối ưu hóa chọn bố cục tốt nhất cho truy vấn
- Tinh chỉnh truy vấn bằng cách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

Tối ưu hóa truy vấn: Đẩy lùi vị ngũ

- Trình tối ưu hóa có thể đẩy các biến vị ngũ phạm vi và đẳng thức xuống thông qua trình kết nối giúp cải thiện hiệu quả lọc
- Engine cung cấp các đầu nối có ràng buộc hai phần:
 - Miền giá trị: phạm vi và tính chất rỗng
 - Vị từ “hộp đen” để lọc

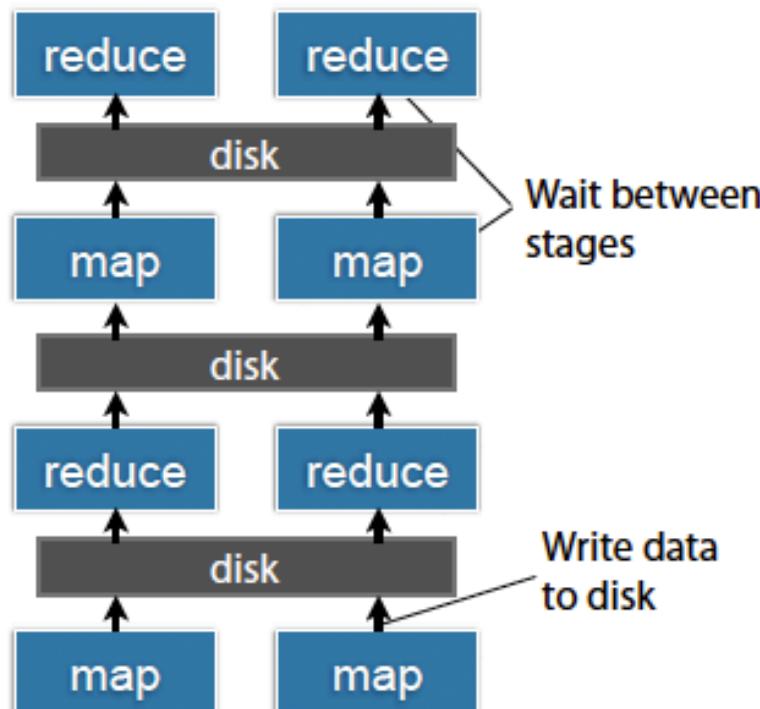
Ví dụ.

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

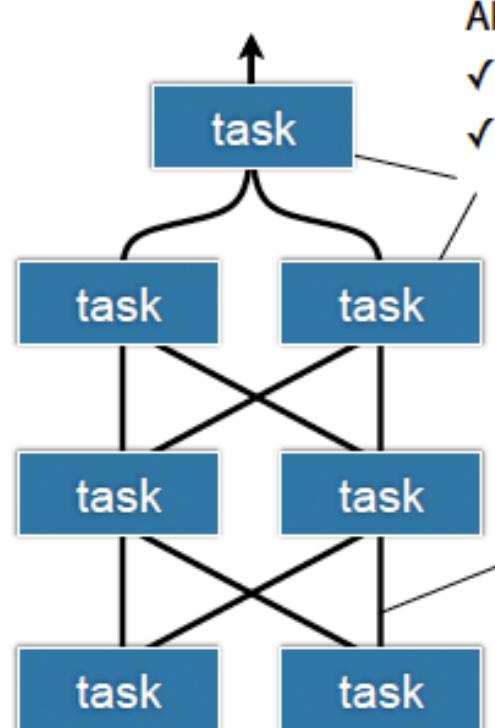


MapReduce vs. Presto

MapReduce



Presto



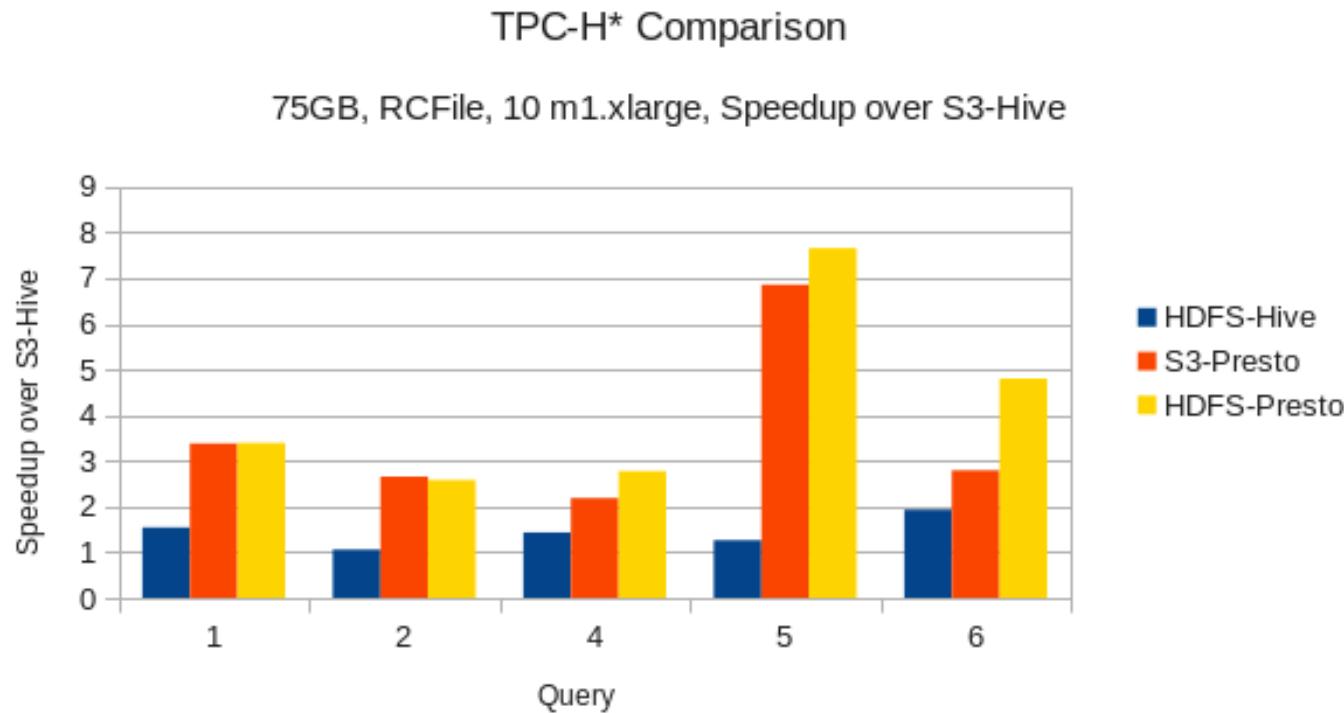
- All stages are pipe-lined
- ✓ No wait time
- ✓ No fault-tolerance

- ✓ No disk IO
- ✓ Data chunk must fit in memory

Thực thi truy vấn

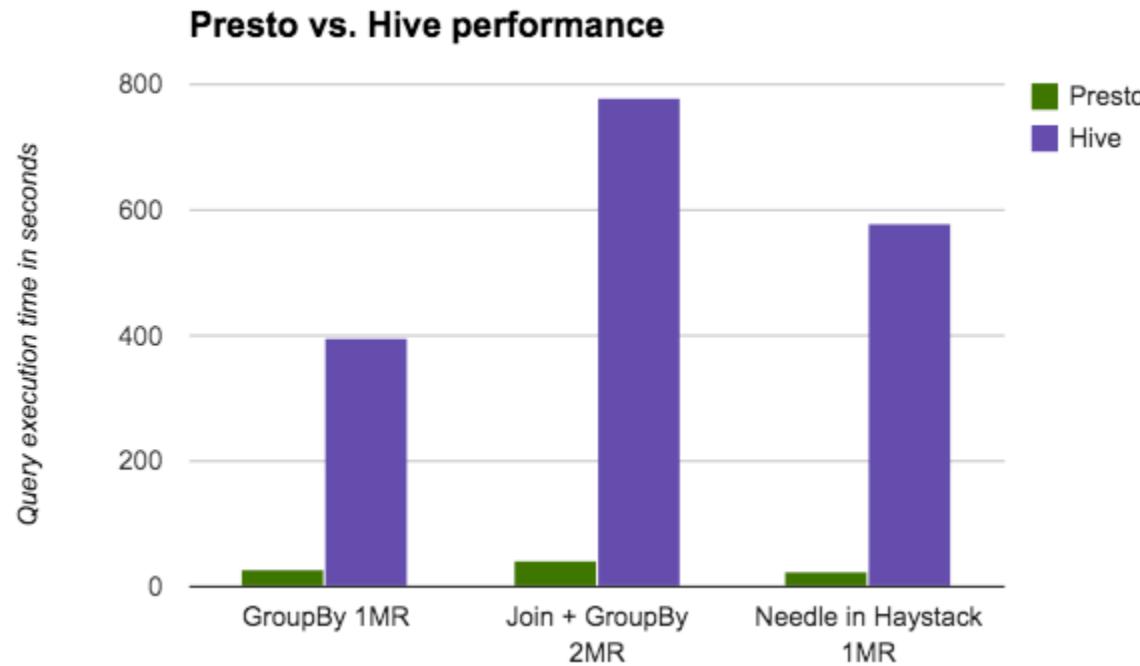
- SQL được chuyển đổi thành các giai đoạn, nhiệm vụ và phân chia
- Mọi tác vụ đều chạy song song
 - Không có thời gian chờ đợi giữa các giai đoạn (đường ống)
 - Nếu một tác vụ thất bại, tất cả các tác vụ đều thất bại cùng một lúc (truy vấn thất bại)
- Truyền dữ liệu từ bộ nhớ này sang bộ nhớ khác
 - 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 nhân viên không chết. Mức tiêu thụ bộ nhớ của tất cả các truy vấn được quản lý hoàn toàn

Điểm chuẩn Qubole Presto

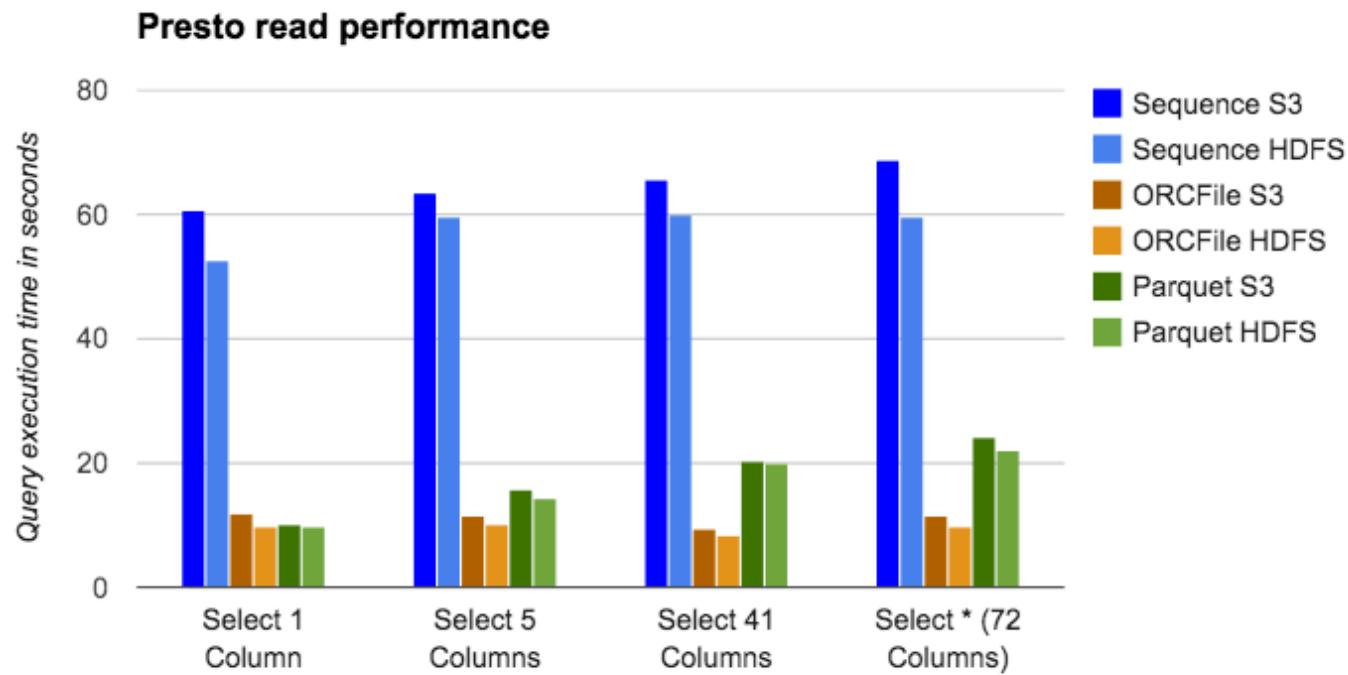


Điểm chuẩn Netflix Presto (1)

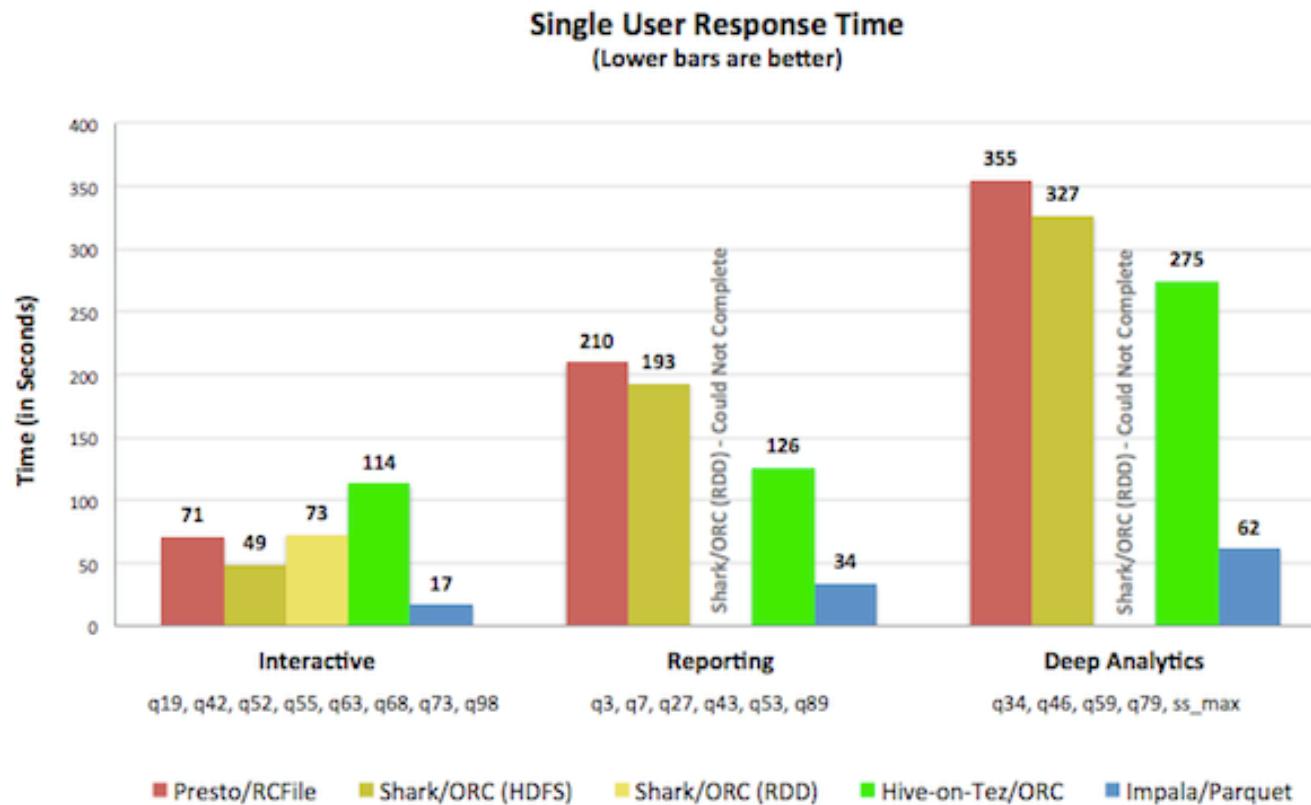
- truy vấn theo nhóm, kết hợp cộng với truy vấn theo nhóm và truy vấn theo nhóm (quét bảng)
- Tệp đầu vào sàn gỗ trên kích thước tệp S3/140GB đến 210GB
- 40 nút m2.4xlarge



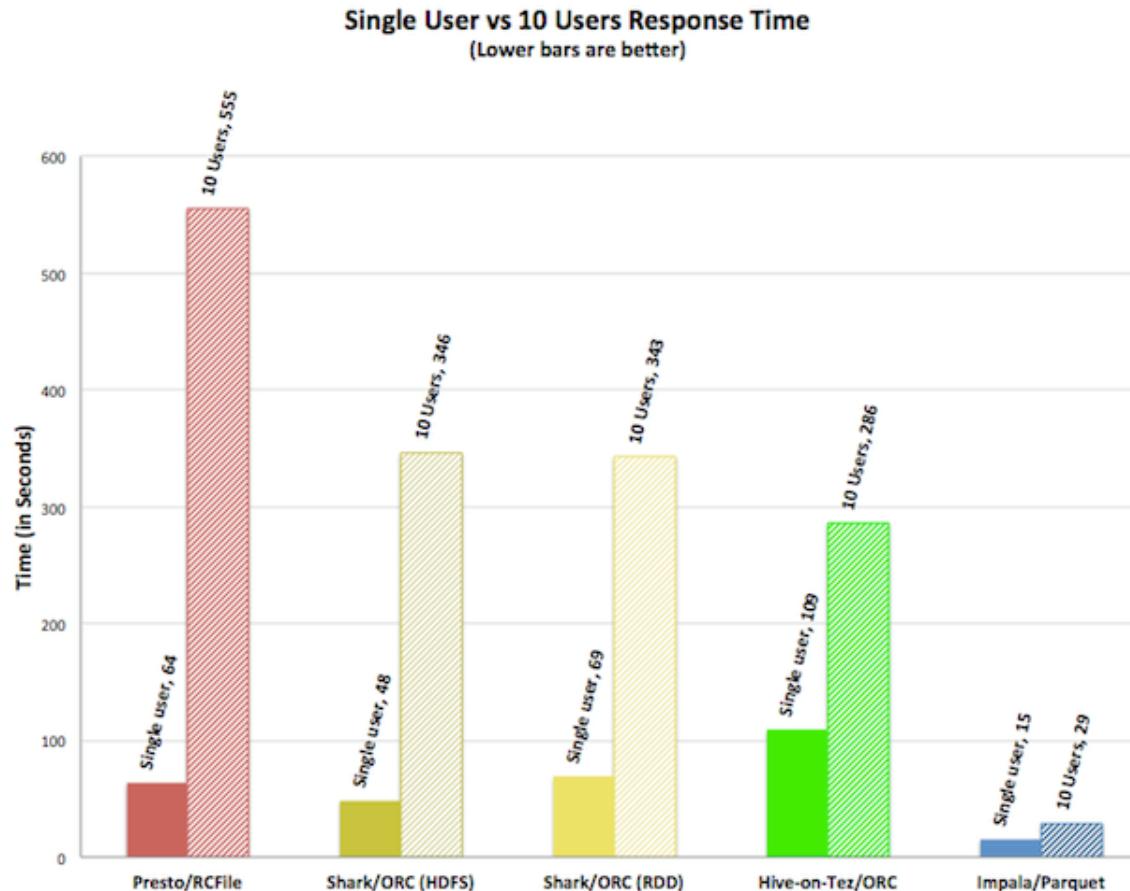
Điểm chuẩn Netflix Presto (2)



Điểm chuẩn Cloudera (1)

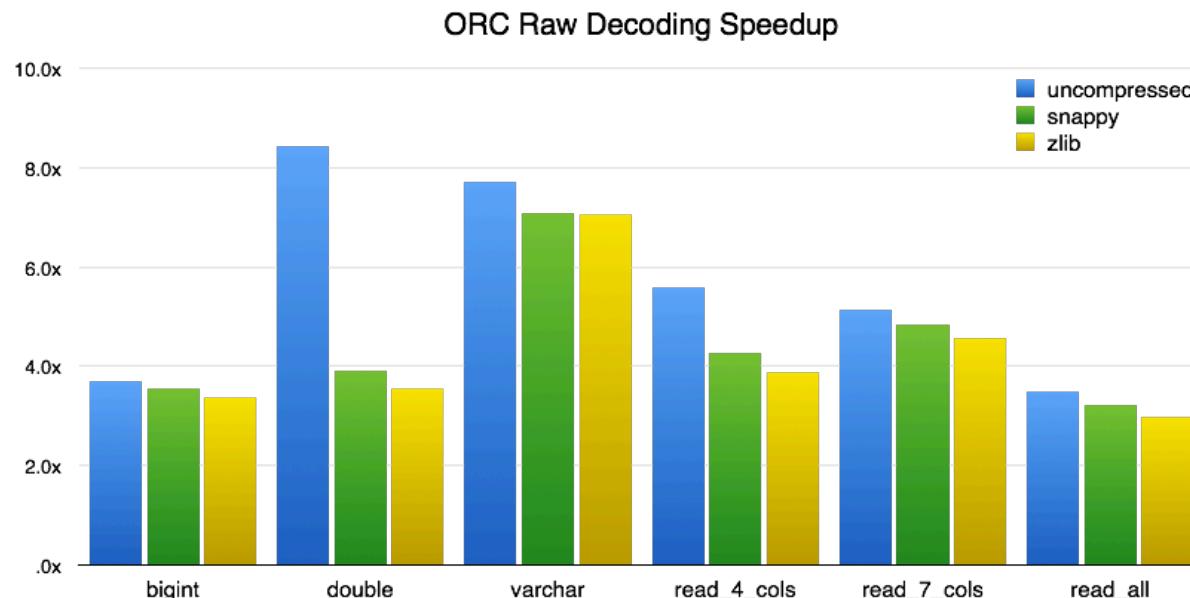


Điểm chuẩn Cloudera (2)



Nhanh hơn nữa: Dữ liệu ở tốc độ của Presto ORC

- 6 triệu hàng sử dụng TPC-H
- Đọc nhiều nhóm cột khác nhau
- Trình đọc ORC dựa trên Hive cũ so với trình đọc ORC Presto mới



Người giới thiệu

- <https://code.facebook.com/posts/370832626374903/enable-faster-data-at-the-speed-of-presto-orc/>
- <https://www.facebook.com/notes/facebookengineering/presto-interactive-with-petabytes-of-data-at-facebook/10151786197628920/>
- Traverso, Martin. "Presto: Tương tác với hàng petabyte dữ liệu tại Facebook." 2014.
- <https://www.slideshare.net/frysuki/presto-hadoopconference-japan-2014>



25
YEARS ANNIVERSARY
SOICT

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

Cảm ơn
cho bạn
chú ý!!!

