Presto

Interactive SQL Query Engine for Big Data

Hadoop Conference in Japan 2014

Sadayuki Furuhashi

Founder & Software Architect
Treasure Data, inc.

TREASURE

A little about me...

- **Sadayuki Furuhashi**
  - github/twitter: @frsyuki
- **Treasure Data, Inc.**
  - Founder & Software Architect
- **Open-source hacker**
  - MessagePack - efficient object serializer
  - Fluentd - data collection tool
  - ServerEngine - Ruby framework to build multiprocess servers
  - LS4 - distributed object storage system
  - kumofs - distributed key-value data store

0. Background + Intro

What's Presto?

A distributed **SQL query engine**
for **interactive** data analisys
against **GBs to PBs of data.**

Presto's history

- **2012 Fall: Project started at Facebook**
  - Designed for interactive query
  - with **speed of commercial data warehouse**
  - and **scalability to the size of Facebook**
- **2013 Winter: Open sourced!**
- **30+ contributes in 6 months**
  - including people from outside of Facebook

What's the problems to solve?

- **We couldn't visualize data in HDFS directly using dashboards or BI tools**
  - because Hive is too slow (not interactive)
  - or ODBC connectivity is unavailable/unstable
- **We needed to store daily-batch results to an interactive DB for quick response (PostgreSQL, Redshift, etc.)**
  - Interactive DB costs more and less scalable by far
- **Some data are not stored in HDFS**
  - We need to copy the data into HDFS to analyze

What's the problems to solve?

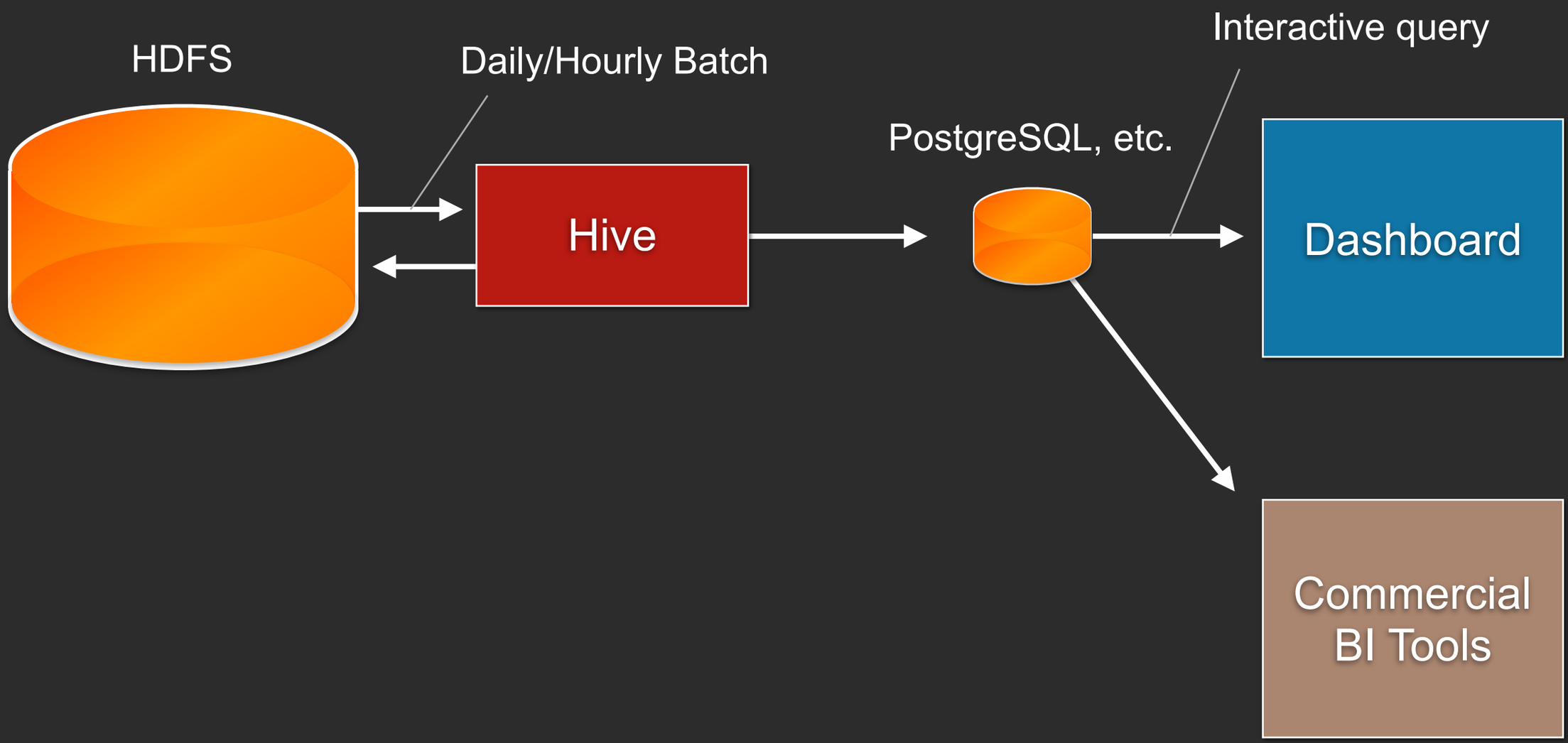
- **We couldn't visualize data in HDFS directly using dashboards or BI tools**
  - because Hive is too slow (not interactive)
  - or ODBC connectivity is unavailable/unstable
- **We needed to store daily-batch results to an interactive DB for quick response (PostgreSQL, Redshift, etc.)**
  - Interactive DB costs more and less scalable by far
- **Some data are not stored in HDFS**
  - We need to copy the data into HDFS to analyze

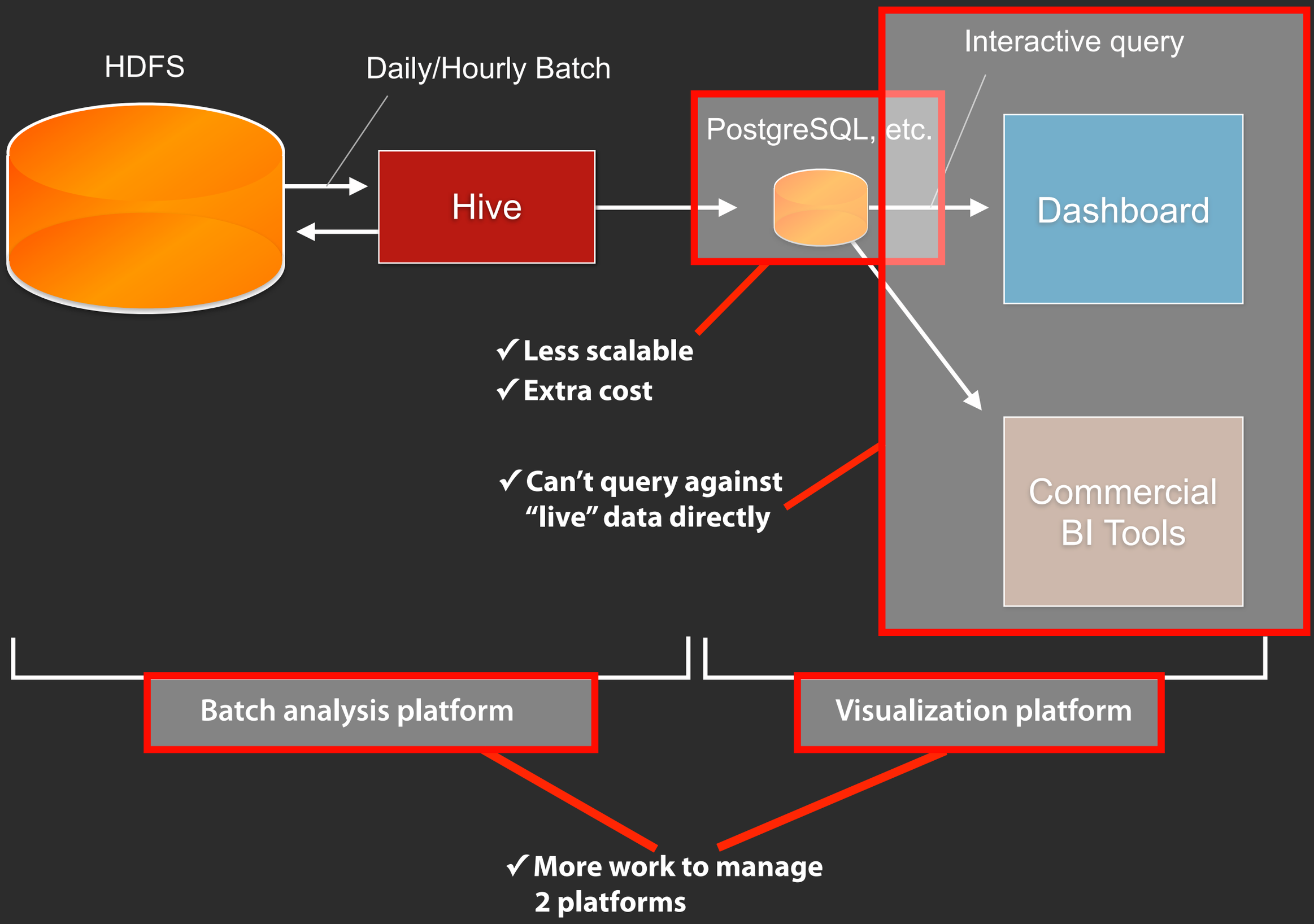
What's the problems to solve?

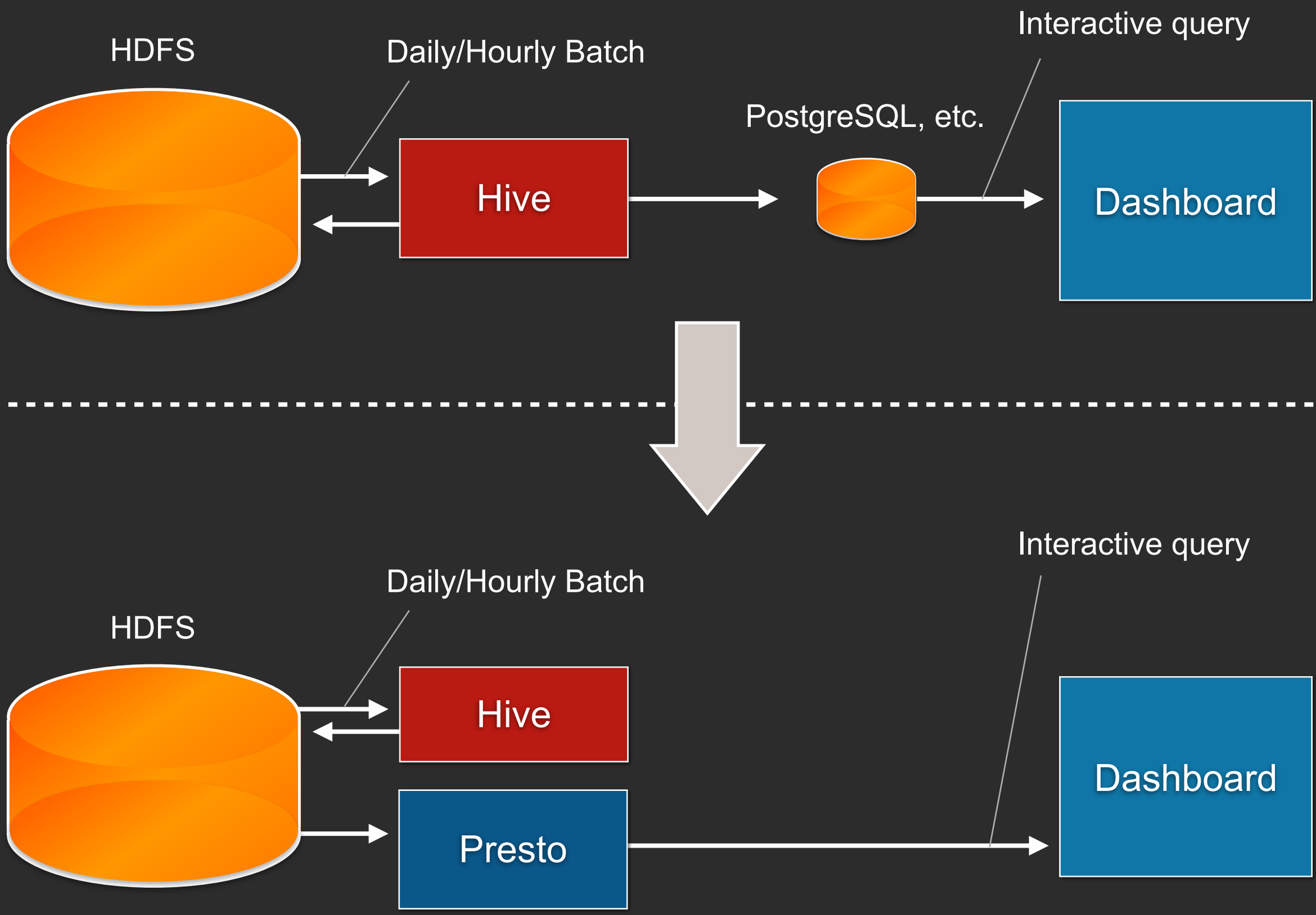
- **We couldn't visualize data in HDFS directly using dashboards or BI tools**
  - because Hive is too slow (not interactive)
  - or ODBC connectivity is unavailable/unstable
- **We needed to store daily-batch results to an interactive DB for quick response (PostgreSQL, Redshift, etc.)**
  - Interactive DB costs more and less scalable by far
- **Some data are not stored in HDFS**
  - We need to copy the data into HDFS to analyze

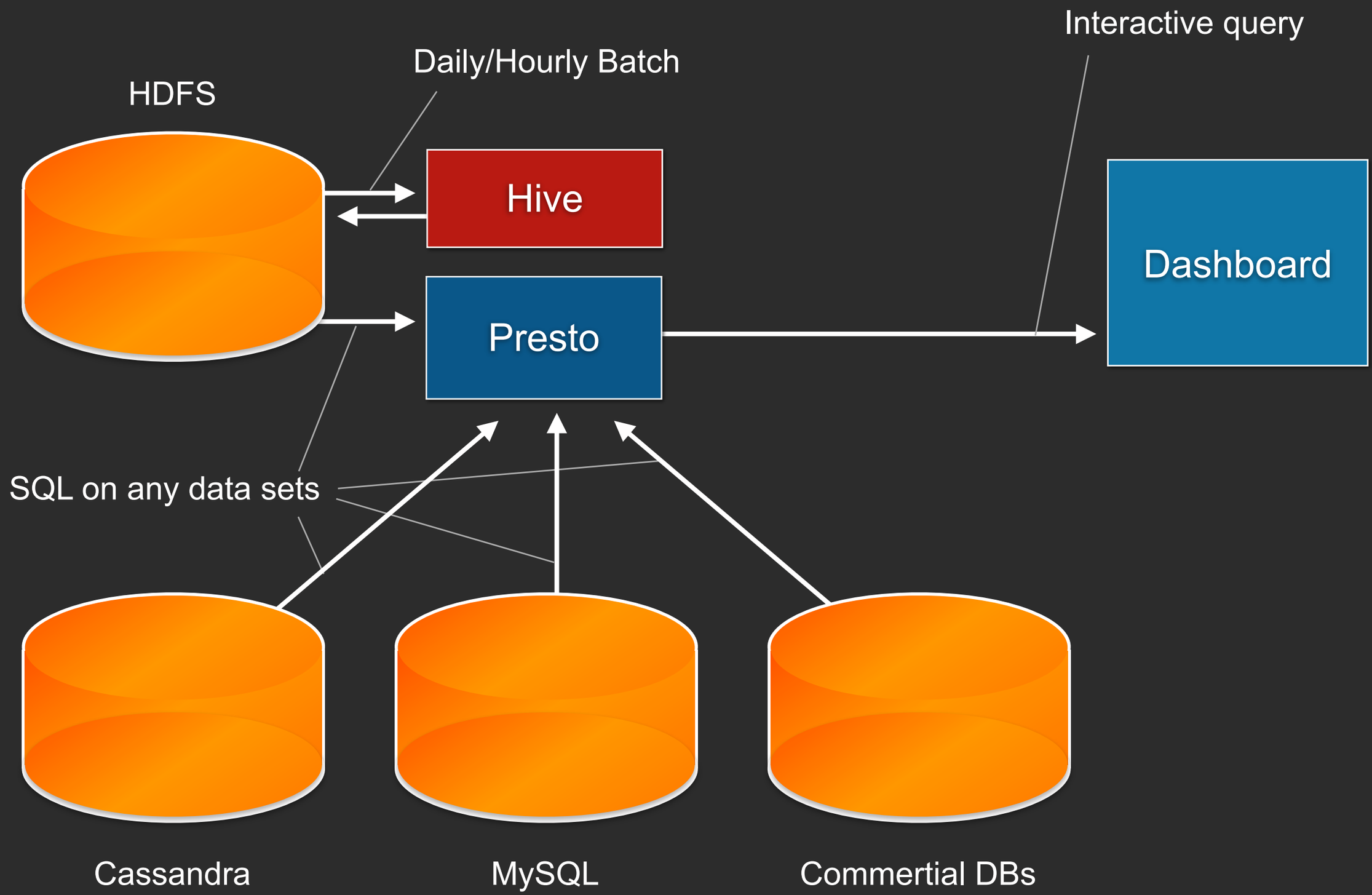
What's the problems to solve?

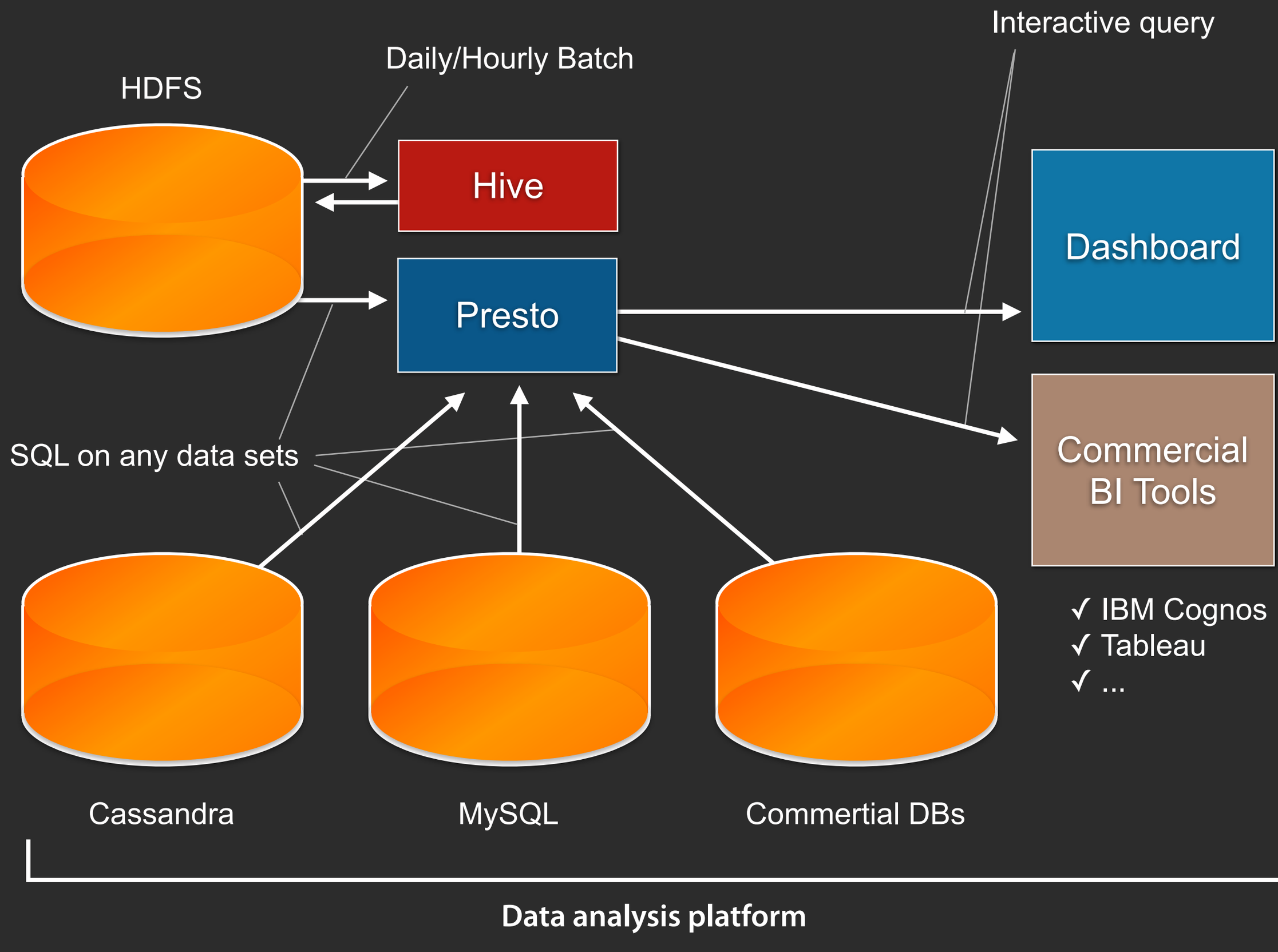
- **We couldn't visualize data in HDFS directly using dashboards or BI tools**
  - because Hive is too slow (not interactive)
  - or ODBC connectivity is unavailable/unstable
- **We needed to store daily-batch results to an interactive DB for quick response (PostgreSQL, Redshift, etc.)**
  - Interactive DB costs more and less scalable by far
- **Some data are not stored in HDFS**
  - We need to copy the data into HDFS to analyze


HDFS
Daily/Hourly Batch
Hive
PostgreSQL, etc.
Interactive query
Dashboard
Commercial
BI Tools
Batch analysis platform
Visualization platform


HDFS
Daily/Hourly Batch
Hive
PostgreSQL, etc.
Interactive query
Dashboard
Commercial BI Tools
✓Less scalable
✓Extra cost
✓Can't query against "live" data directly
Batch analysis platform
Visualization platform
✓More work to manage 2 platforms


HDFS
Daily/Hourly Batch
Hive
PostgreSQL, etc.
Interactive query
Dashboard
HDFS
Daily/Hourly Batch
Hive
Presto
Interactive query
Dashboard


Interactive query
Daily/Hourly Batch
HDFS
Hive
Presto
Dashboard
SQL on any data sets
Cassandra
MySQL
Commertial DBs


Interactive query
Daily/Hourly Batch
HDFS
Hive
Presto
Dashboard
SQL on any data sets
Commercial
BI Tools
✓ IBM Cognos
✓ Tableau
✓ ...
Cassandra
MySQL
Commertial DBs
Data analysis platform

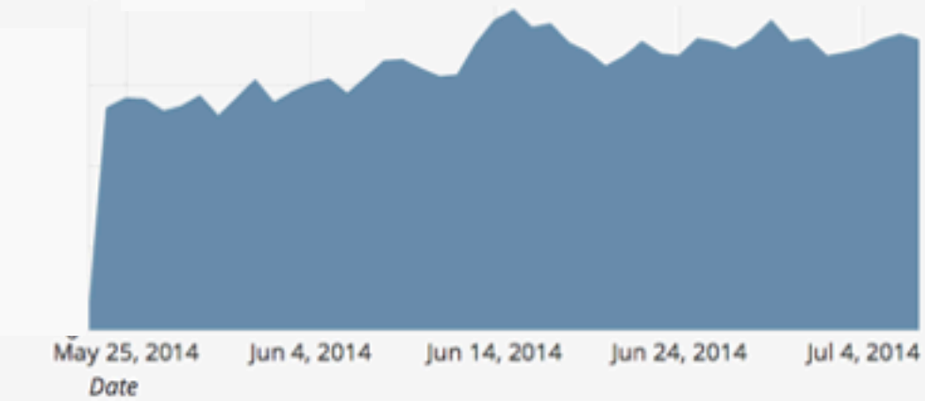
Treasure Data

Dashboards Settings Help

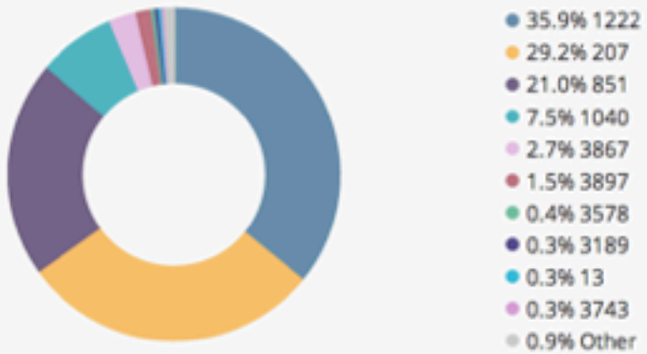
Example Presto Dashboard

Add Element Arrange Presen

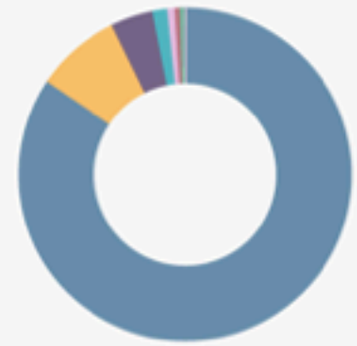
Daily



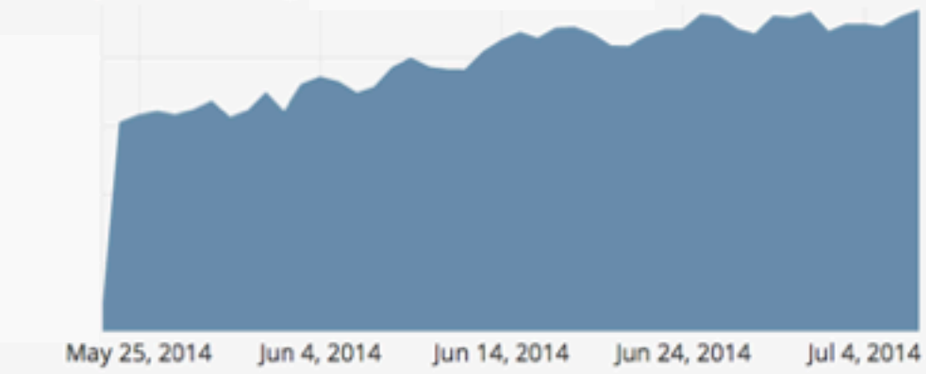
Query Ratio



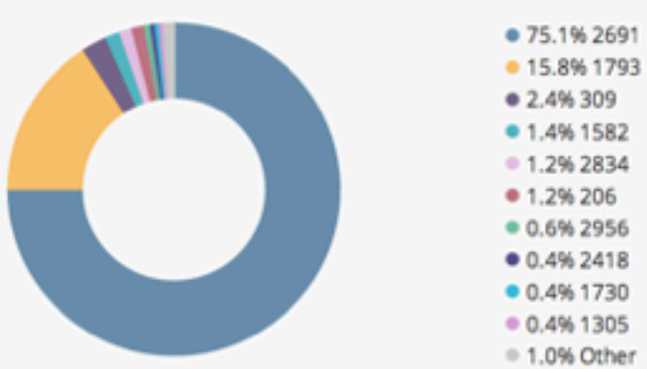
Query Ratio



Daily Streaming

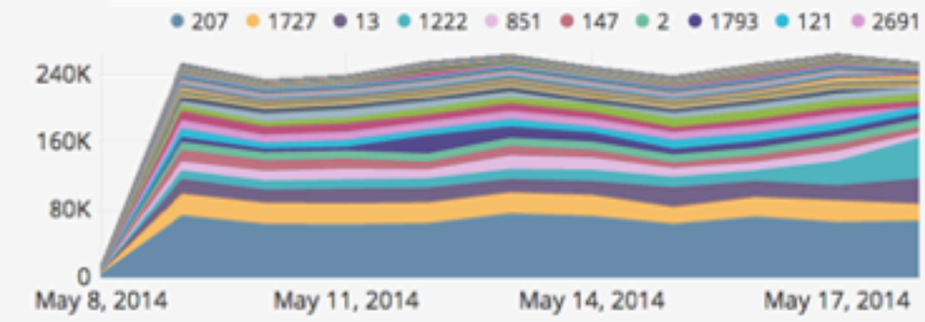


Job

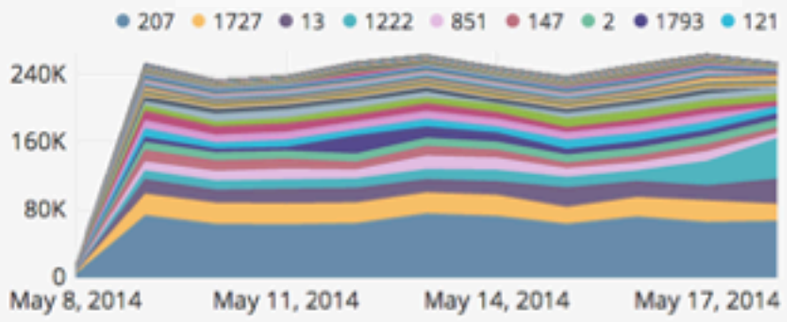


Import Counts

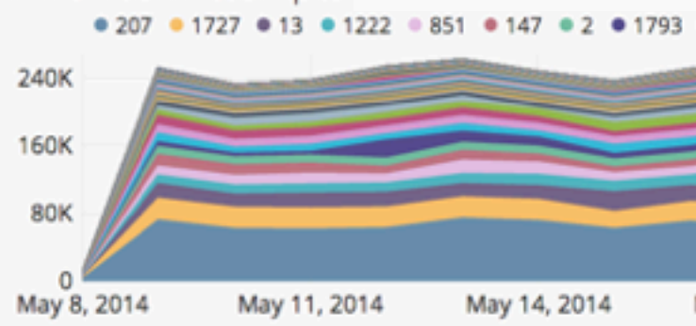
Hive



Hive Task



Hive Task Attempts



dashboard on chart.io: https://chartio.com/

What can Presto do?

- Query **interactively** (in milli-seconds to minues)
  - MapReduce and Hive are still necessary for ETL
- Query using **commercial BI tools** or dashboards
  - Reliable ODBC/JDBC connectivity
- Query across **multiple data sources** such as Hive, HBase, Cassandra, or even commertial DBs
  - Plugin mechanism
- Integrate batch analisys + visualization into a single data analysis platform

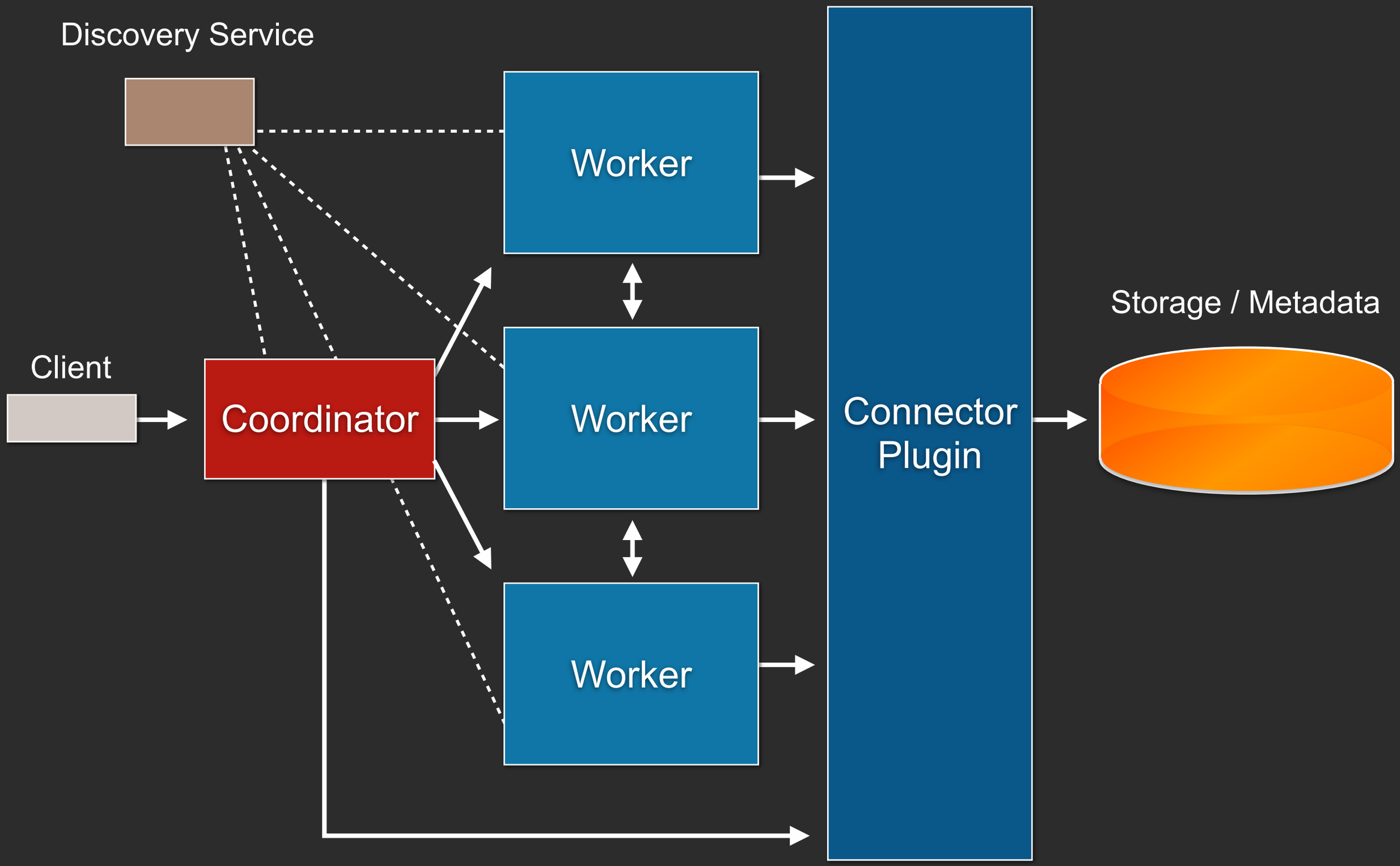
Presto's deployment

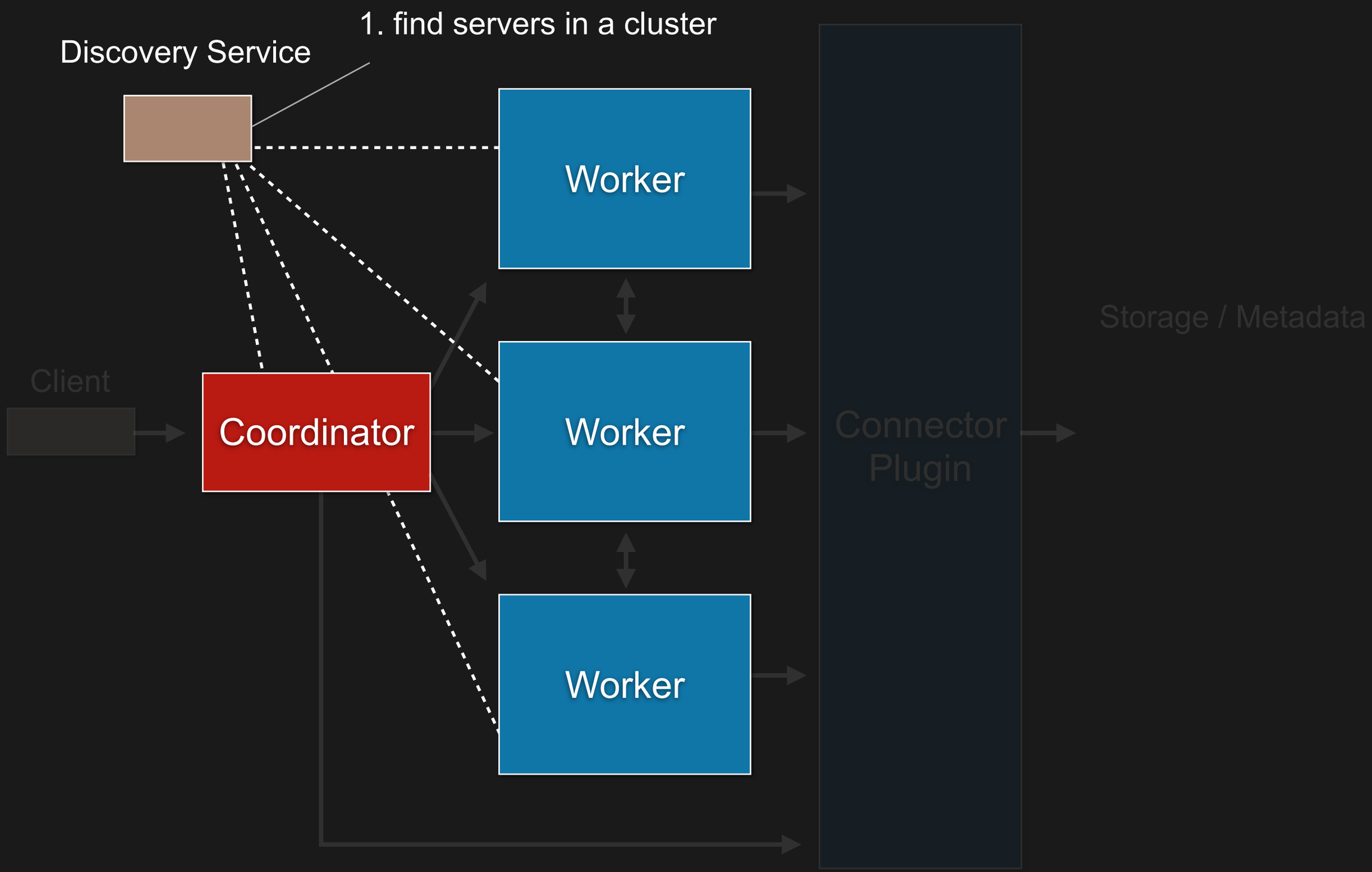
- **Facebook**
  - Multiple geographical regions
  - scaled to 1,000 nodes
  - actively used by 1,000+ employees
  - who run 30,000+ queries every day
  - processing 1PB/day
- **Netflix, Dropbox, Treasure Data, Airbnb, Qubole**
- **Presto as a Service**

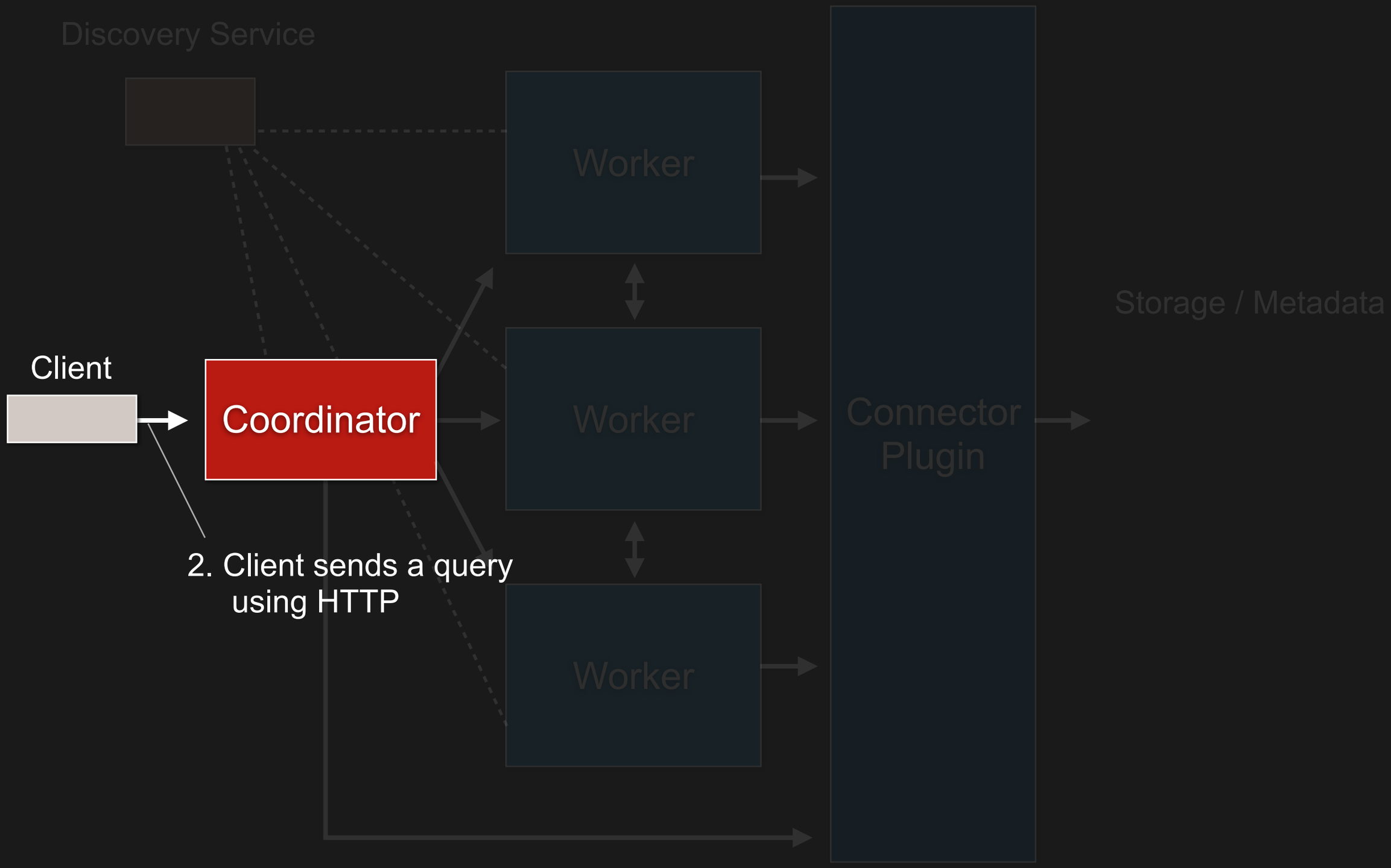
Today's talk

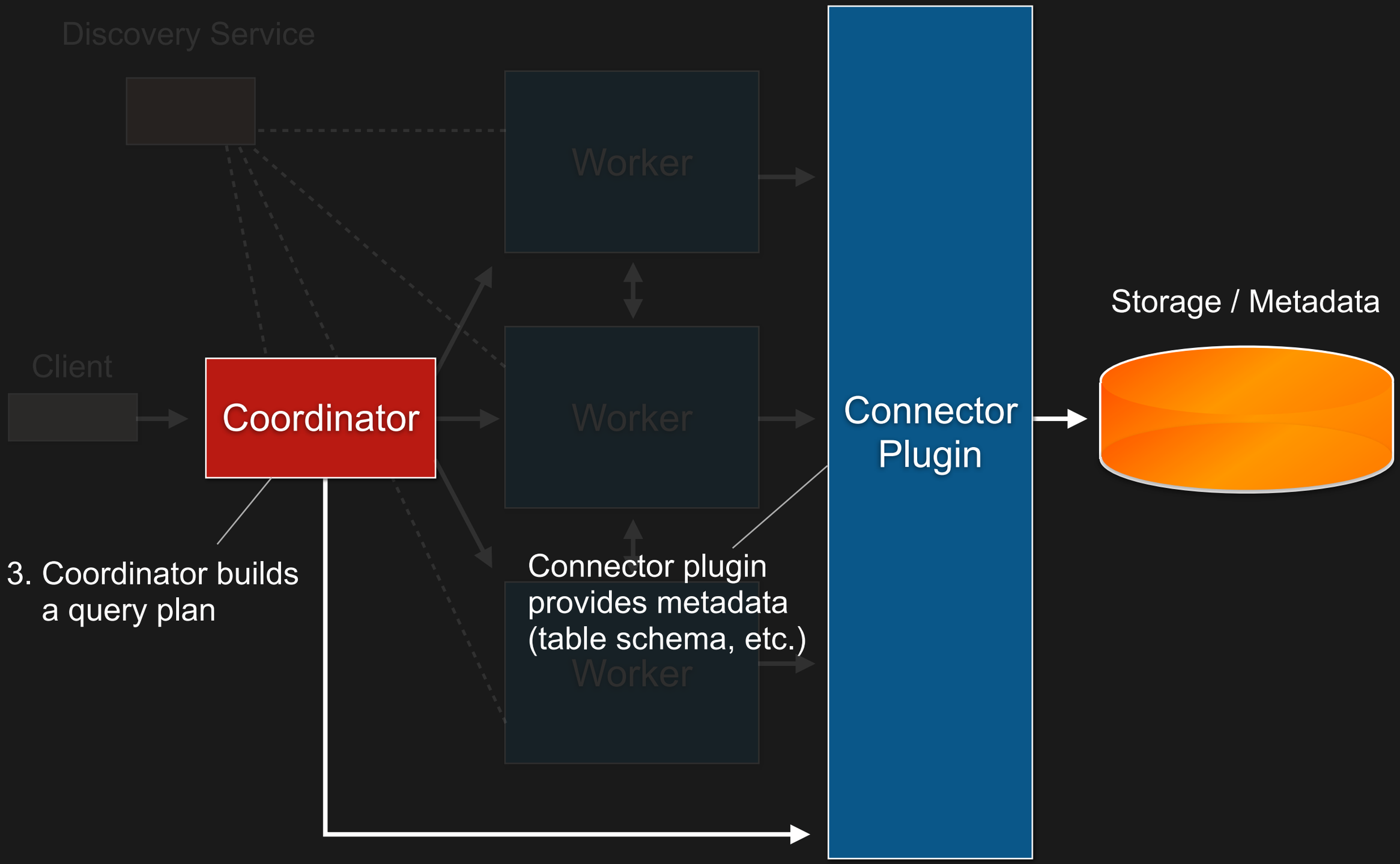
1. Distributed architecture
2. Data visualization - Demo
3. Query Execution - Presto vs. MapReduce
4. Monitoring & Configuration
5. Roadmap - the future

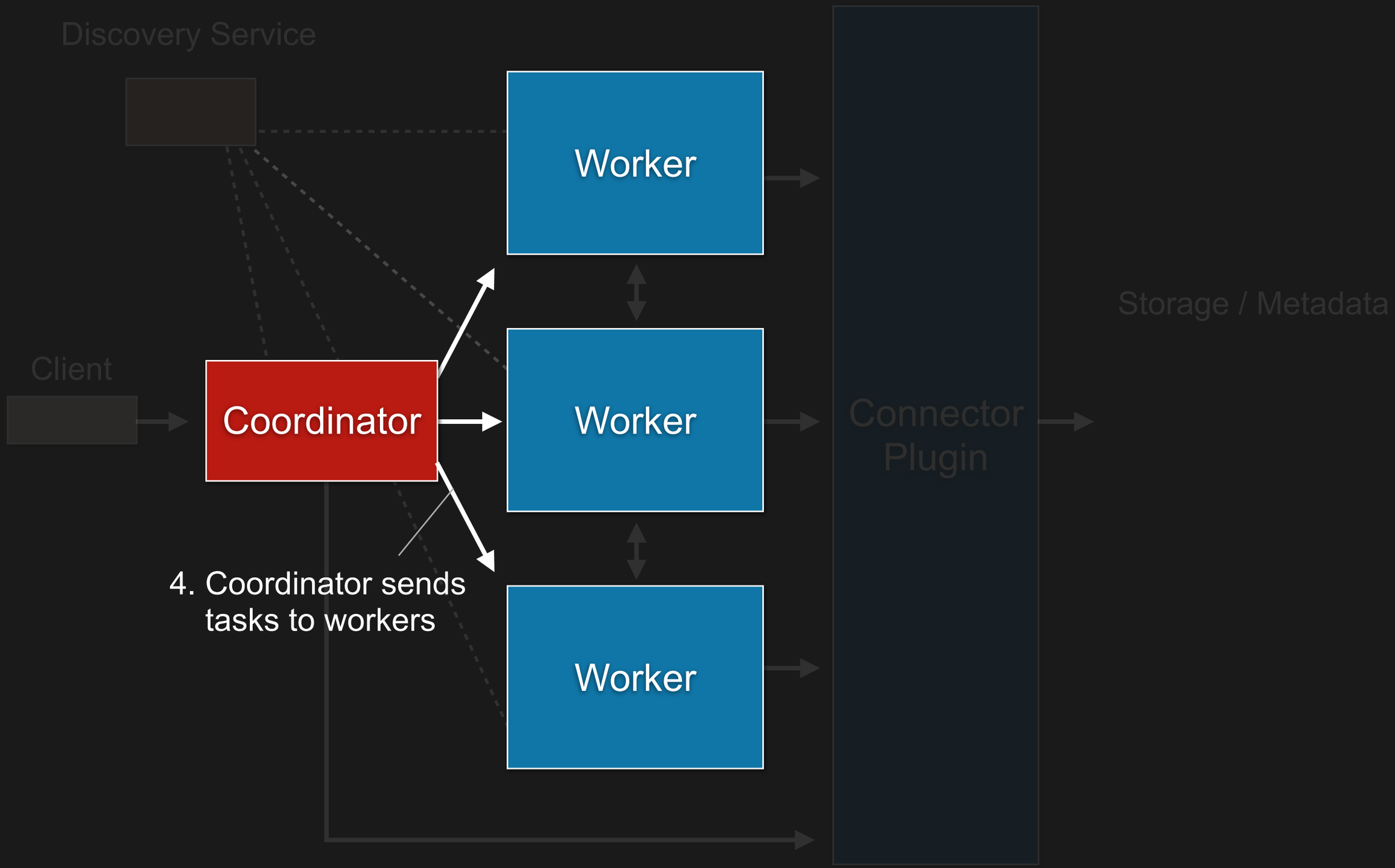
1. Distributed architecture

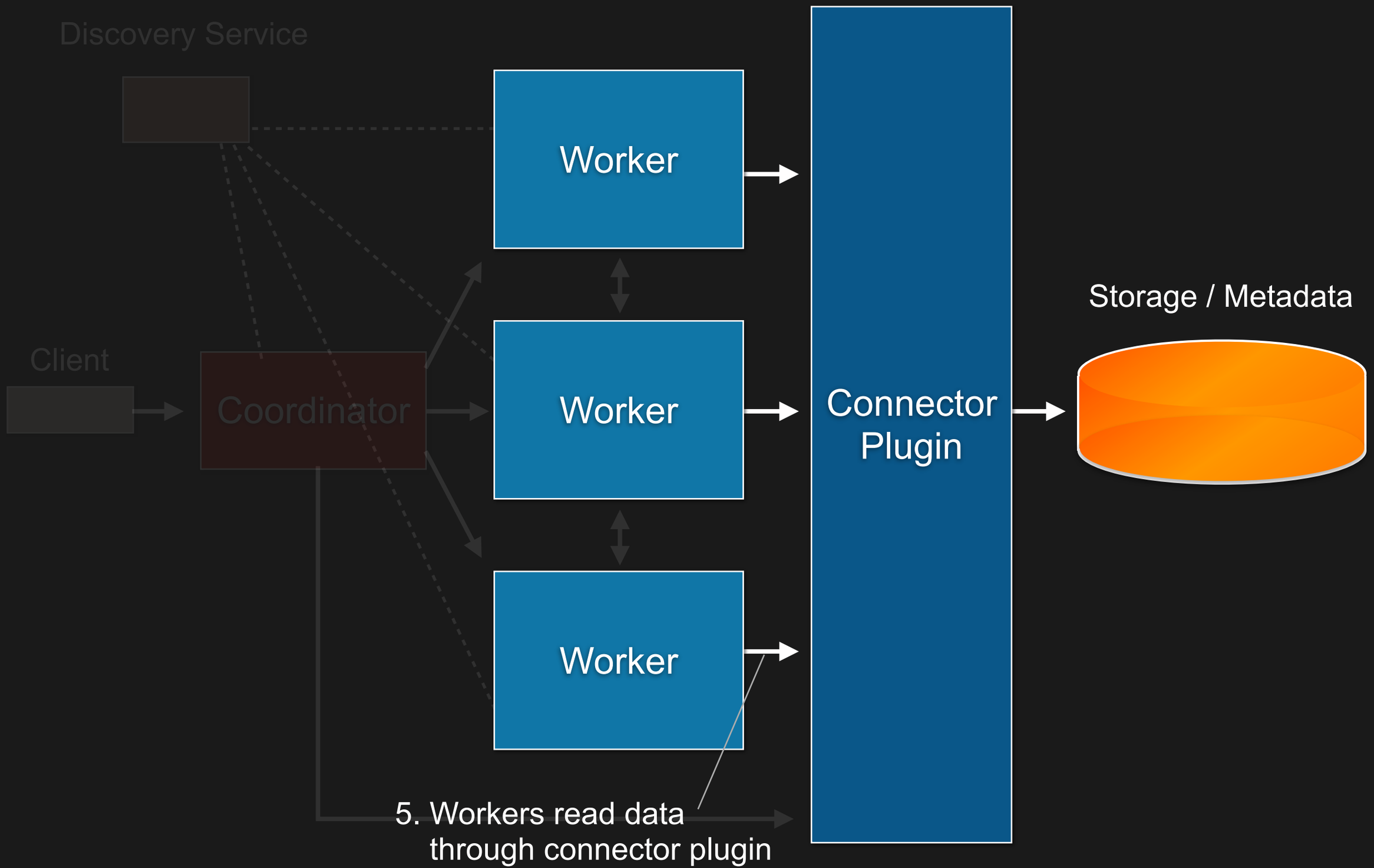

Discovery Service
Client
Coordinator
Worker
Worker
Worker
Connector Plugin
Storage / Metadata


1. find servers in a cluster
Discovery Service
Worker
Storage / Metadata
Client
Coordinator
Worker
Connector
Plugin
Worker


Discovery Service
Worker
Storage / Metadata
Client
Coordinator
Worker
Connector Plugin
Worker
2. Client sends a query using HTTP


Discovery Service
Worker
Client
Coordinator
Worker
Connector Plugin
Storage / Metadata
3. Coordinator builds a query plan
Connector plugin provides metadata (table schema, etc.)
Worker


Discovery Service
Worker
Storage / Metadata
Client
Coordinator
Worker
Connector
Plugin
4. Coordinator sends
tasks to workers
Worker


Discovery Service
Client
Coordinator
Worker
Worker
Worker
Connector Plugin
Storage / Metadata
5. Workers read data through connector plugin

Discovery Service

Client

Coordinator

Worker

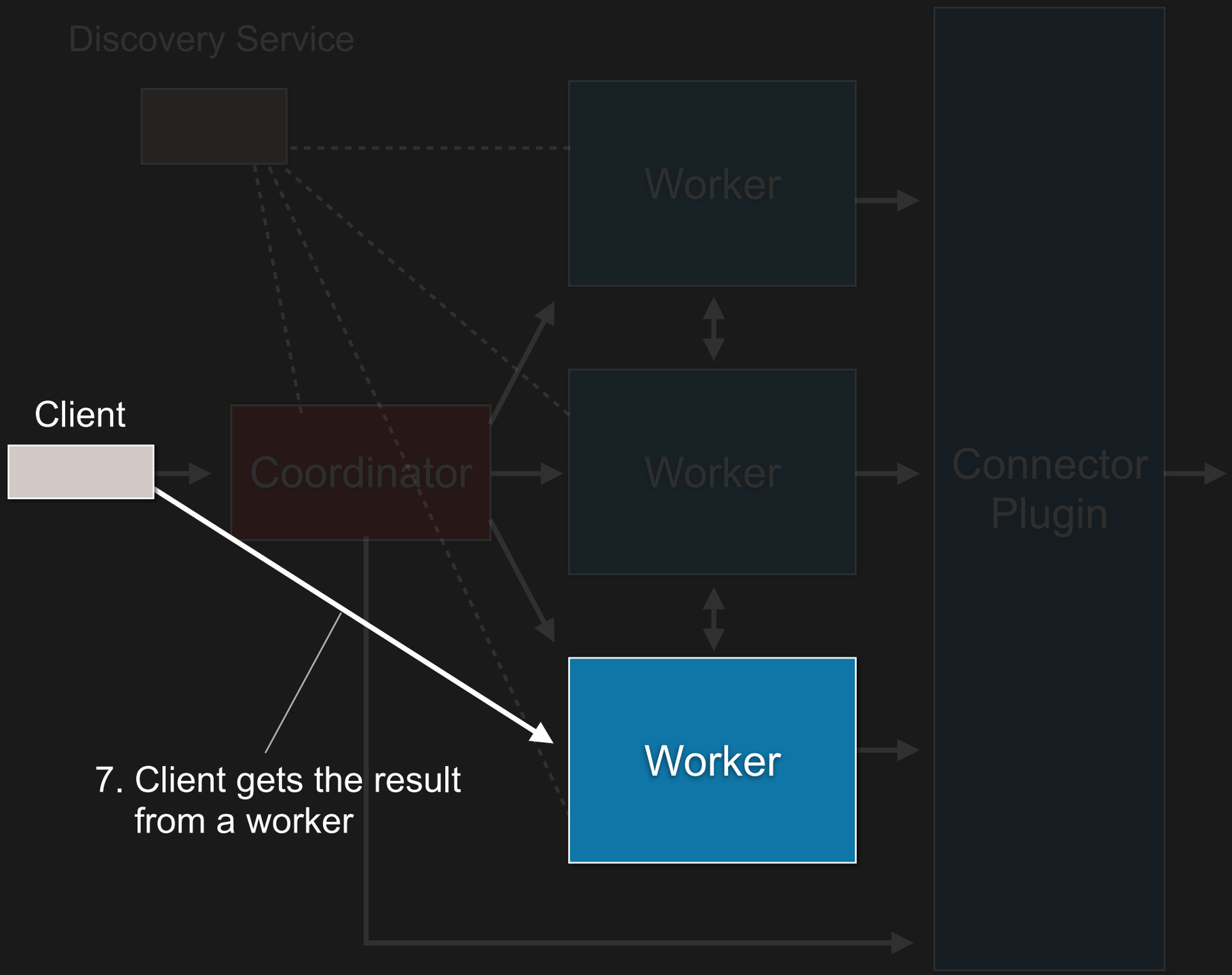
Worker

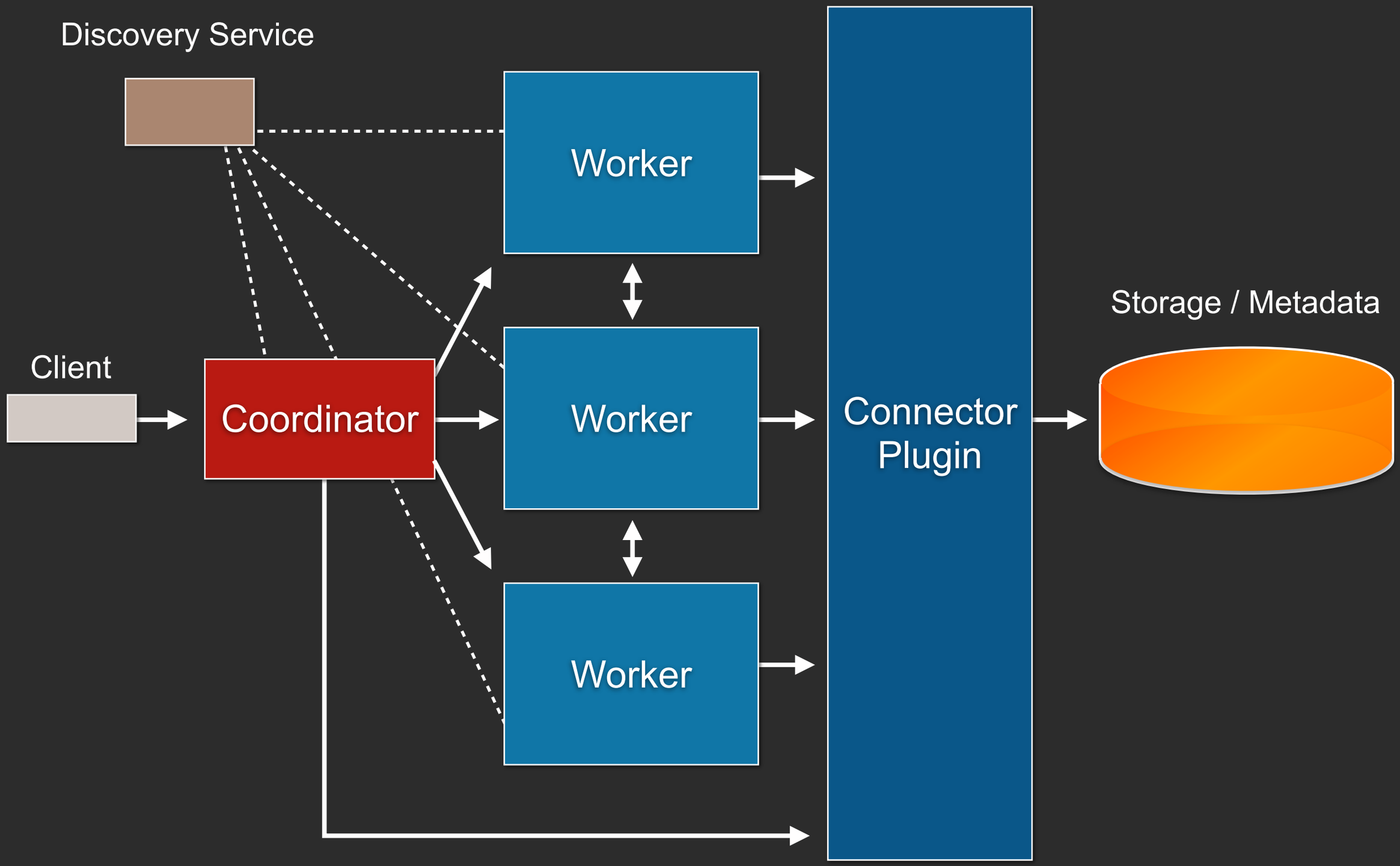
Worker

Connector Plugin

Storage / Metadata

6. Workers run tasks in memory

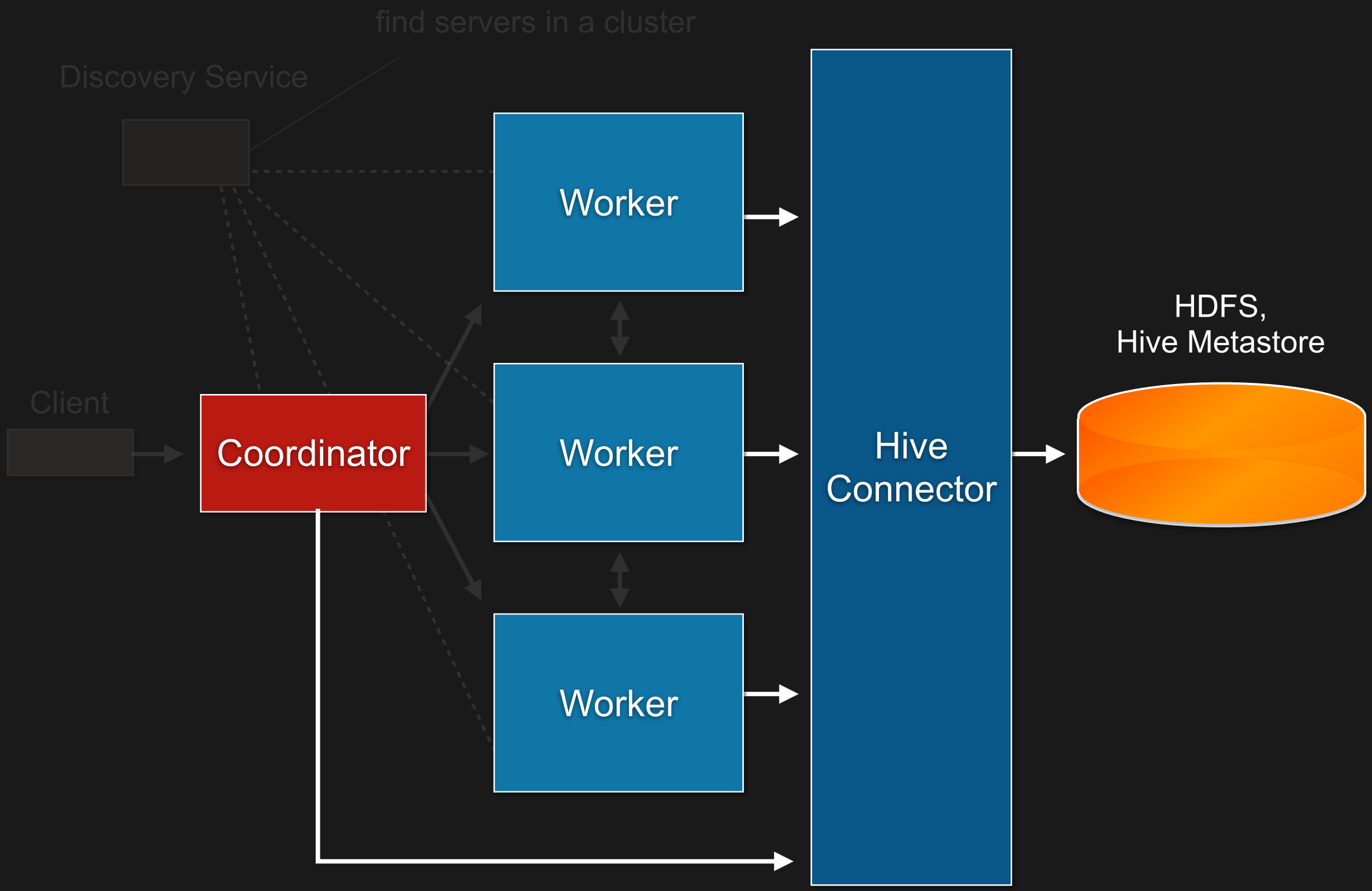

Discovery Service
Worker
Storage / Metadata
Client
Coordinator
Worker
Connector Plugin
Worker
7. Client gets the result from a worker


Discovery Service
Client
Coordinator
Worker
Worker
Worker
Connector Plugin
Storage / Metadata

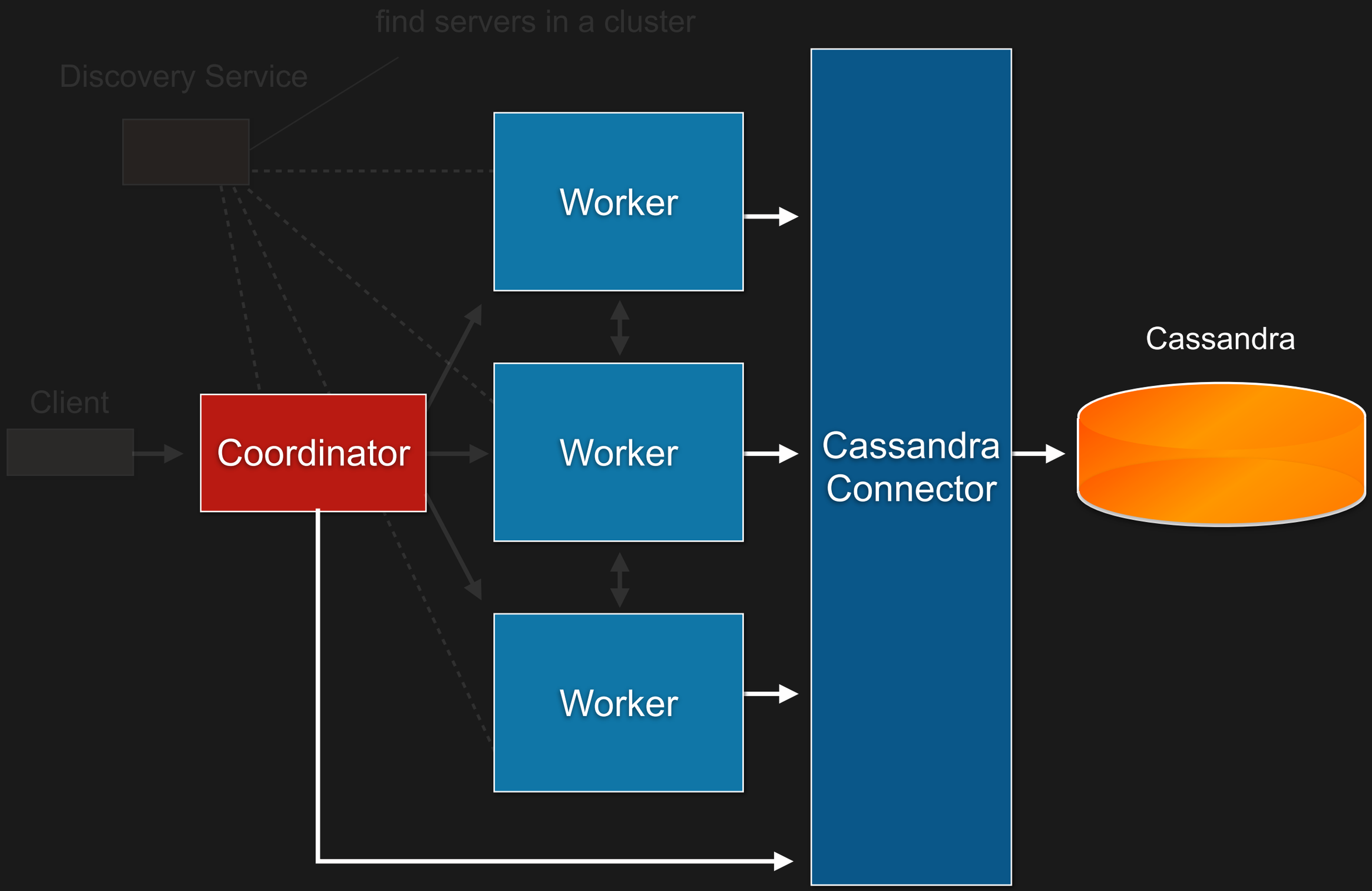
What's Connectors?

- **Connectors are plugins to Presto**
  - written in Java
- **Access to storage and metadata**
  - provide table schema to coordinators
  - provide table rows to workers
- **Implementations:**
  - Hive connector
  - Cassandra connector
  - MySQL through JDBC connector (prerelease)
  - Or your own connector

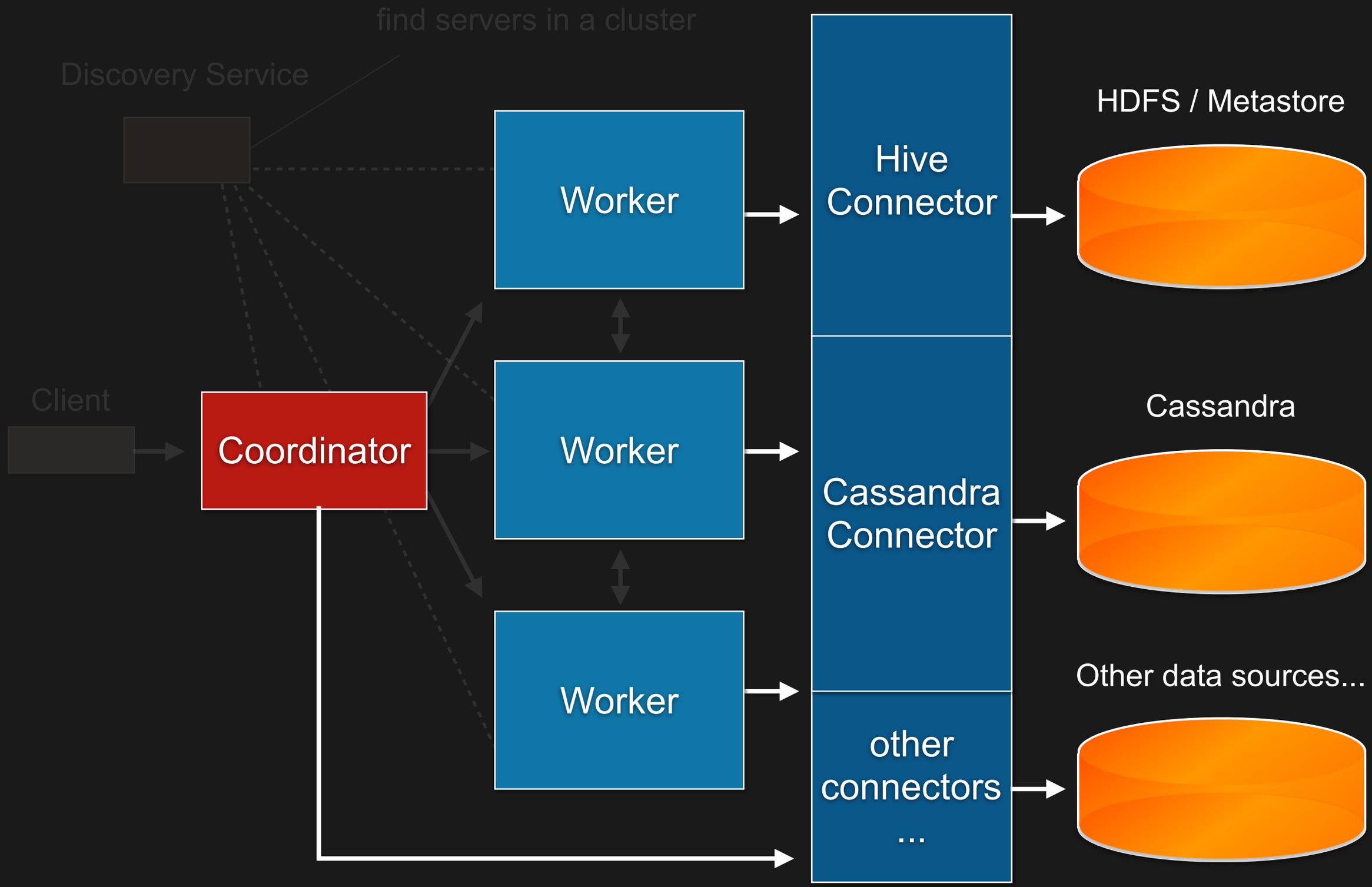
Hive connector



Cassandra connector



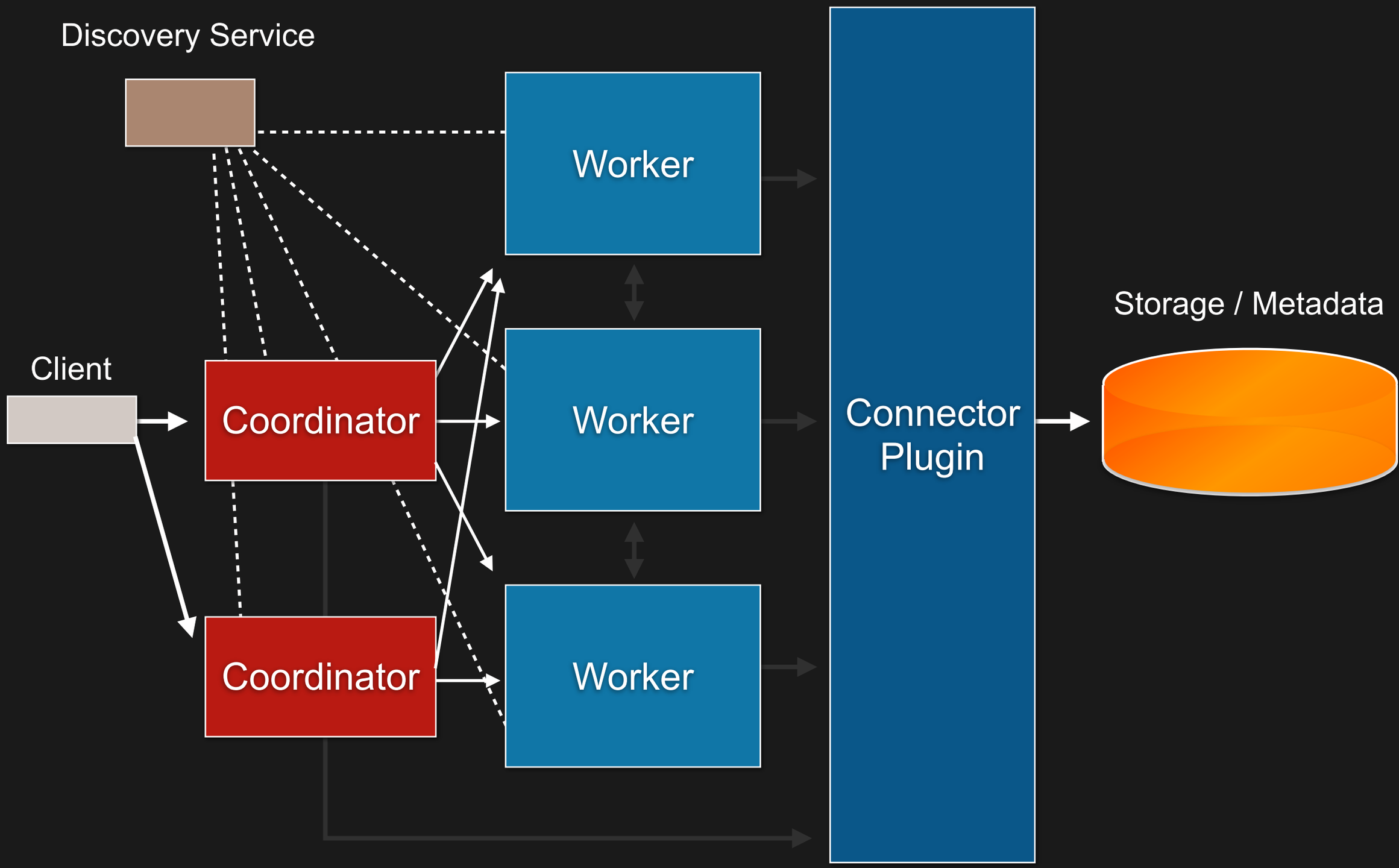
Multiple connectors in a query



1. Distributed architecture

- **3 type of servers:**
  - Coordinator, worker, discovery service
- **Get data/metadata through connector plugins.**
  - Presto is NOT a database
  - Presto provides SQL to existent data stores
- **Client protocol is HTTP + JSON**
  - Language bindings:
    Ruby, Python, PHP, Java (JDBC), R, Node.JS...

Coordinator HA



2. Data visualization

The problems to use BI tools

- **BI tools need ODBC or JDBC connectivity**
  - Tableau, IBM Cognos, QlickView, Chart.IO, ...
  - JasperSoft, Pentaho, MotionBoard, ...
- **ODBC/JDBC is VERY COMPLICATED**
  - Matured implementation needs LONG time

A solution: PostgreSQL protocol

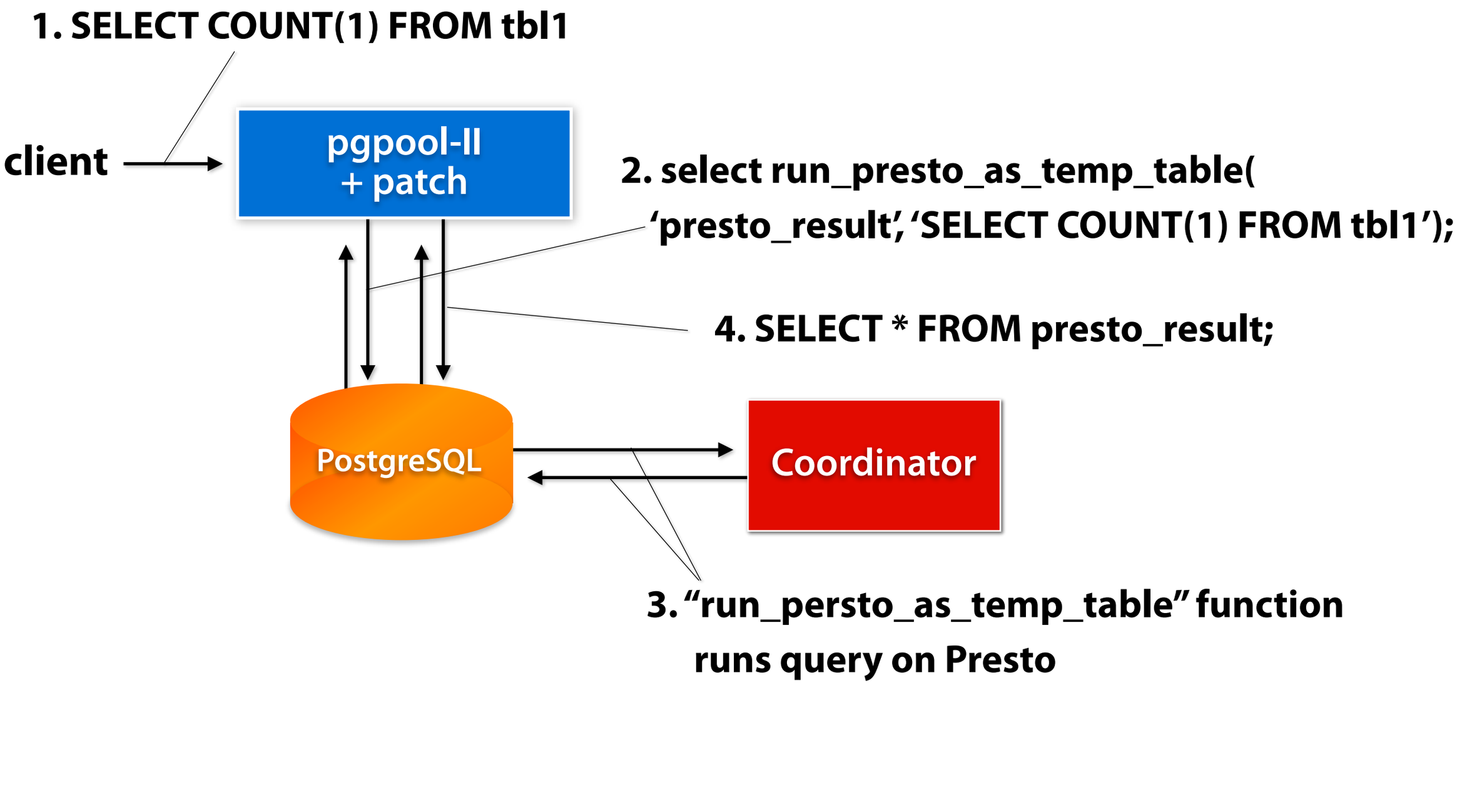
- Creating a PostgreSQL protocol **gateway**
- Using PostgreSQL's **stable** ODBC / JDBC driver

Prestogres

PostgreSQL protocol gateway for Presto

https://github.com/treasure-data/prestogres

How Prestogres works?



Demo

2. Data visualization with Presto

- **Data visualization tools need ODBC/JDBC driver**
  - but implemetation takes LONG time
- **A solution is to use PostgreSQL protocol**
  - and use PostgreSQL's ODBC/JDBC driver
- **Prestogres is already confirmed to work with some commertial BI tools**

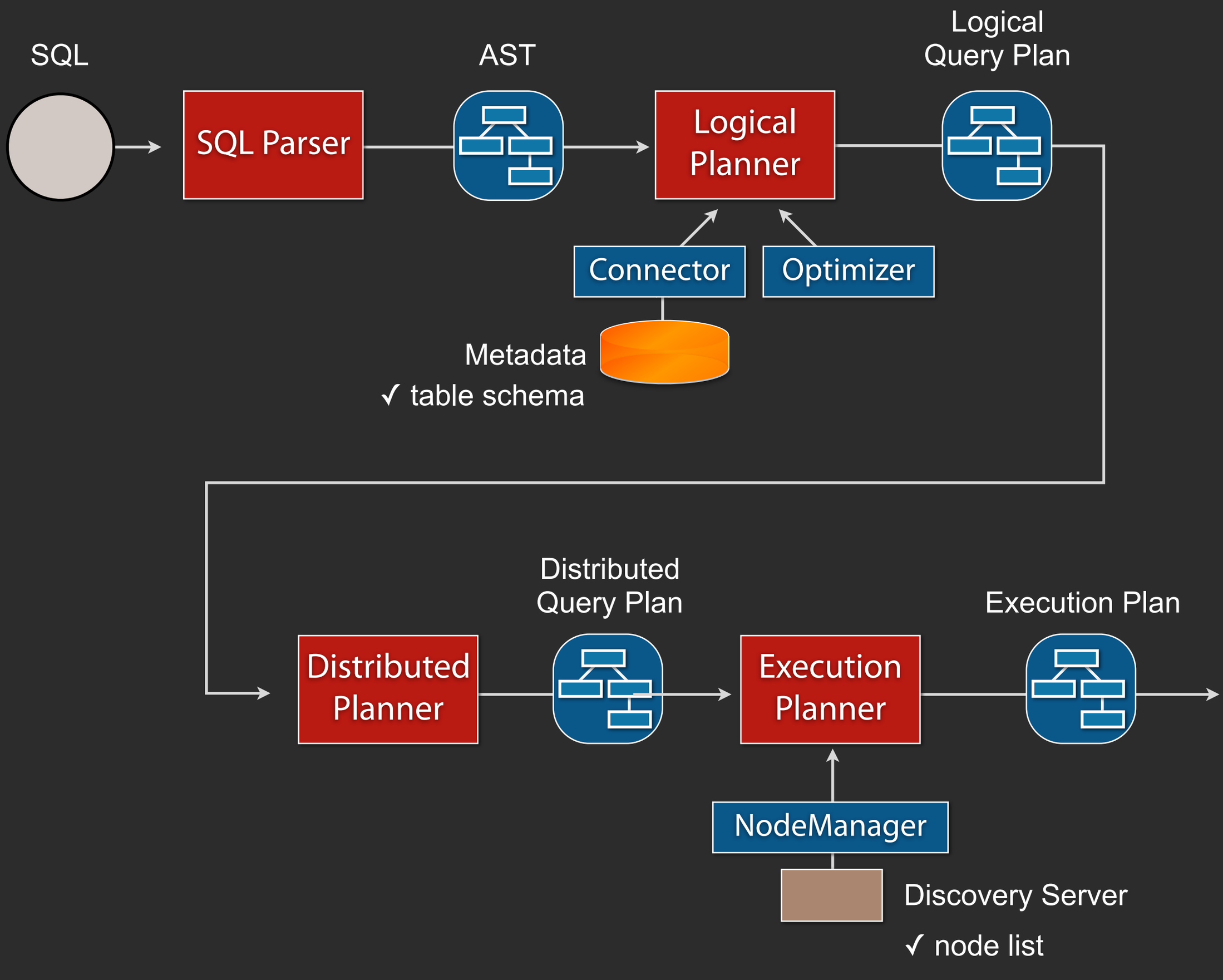
3. Query Execution

Presto's execution model

- **Presto is NOT MapReduce**
- **Presto's query plan is based on DAG**
  - more like Apache Tez or traditional MPP databases

How query runs?

- **Coordinator**
  - SQL Parser
  - Query Planner
  - Execution planner
- **Workers**
  - Task execution scheduler


SQL
AST
Logical
Query Plan
SQL Parser
Logical
Planner
Connector
Optimizer
Metadata
✓ table schema
Distributed
Query Plan
Execution Plan
Distributed
Planner
Execution
Planner
NodeManager
Discovery Server
✓ node list

(today's talk)

SQL

SQL Parser

Logical Query Plan

Query Planner

Connector

Optimizer

Metadata

✓ table schema

Query Plan

Distributed Planner

Execution Planner

Execution Plan

NodeManager

Discovery Service

✓ node list

Query Planner

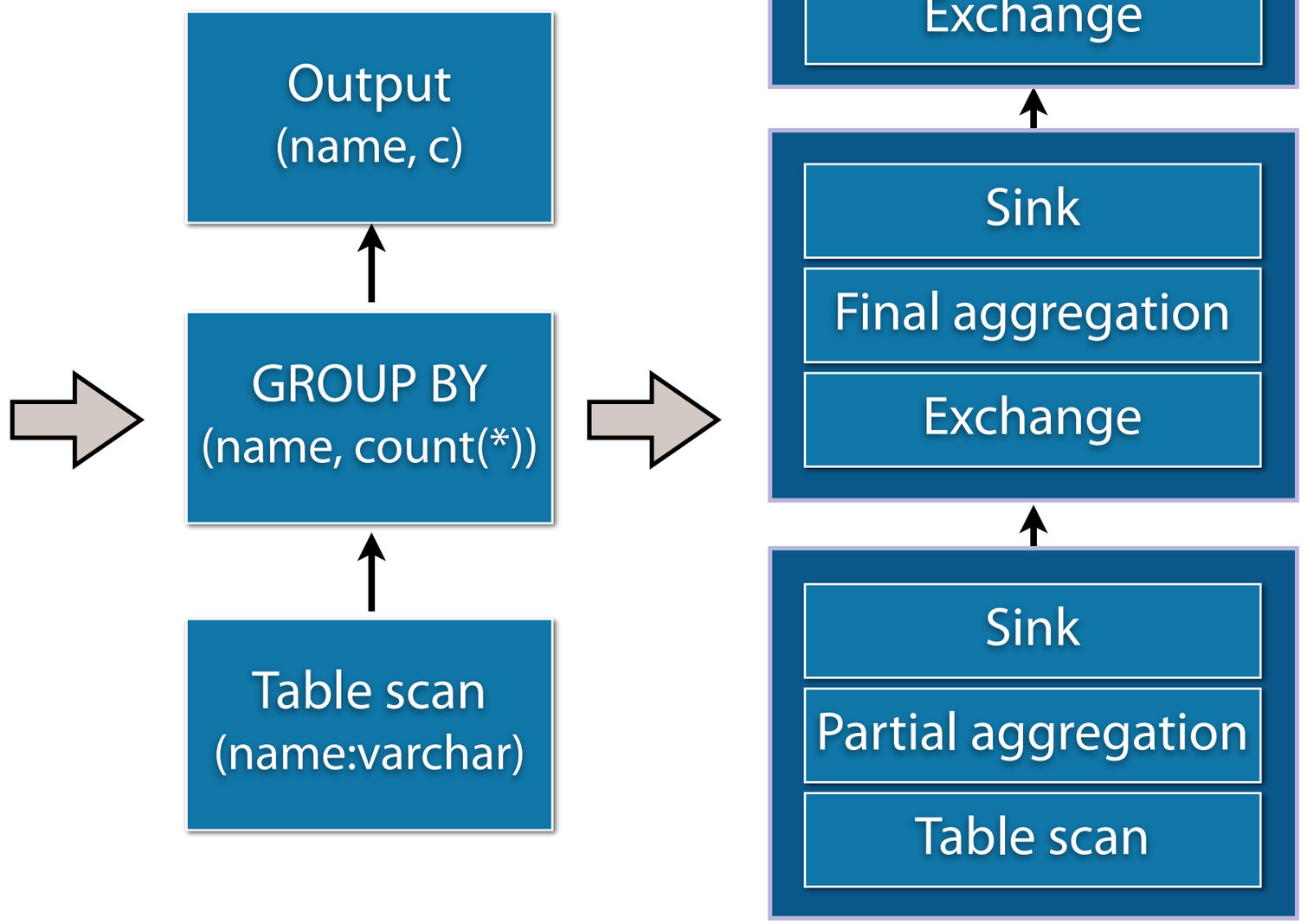
SQL

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

+

Table schema

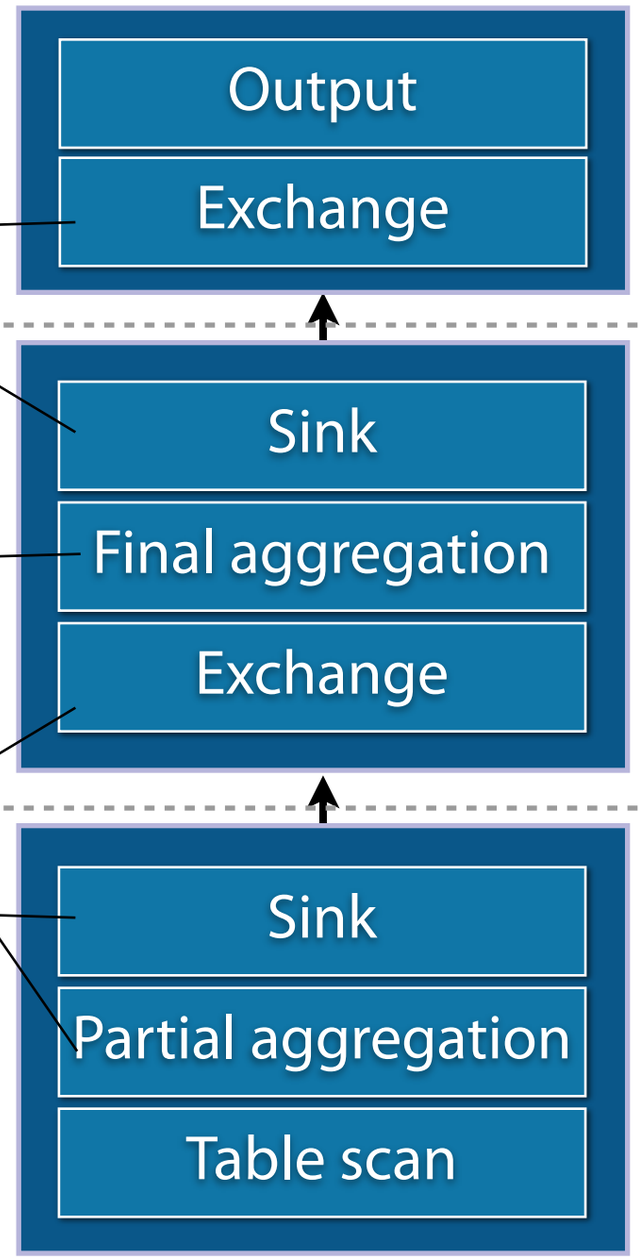
```
impressions (
   name varchar
   time bigint
)
```



Query Planner - Stages

Stage-0

inter-worker data transfer



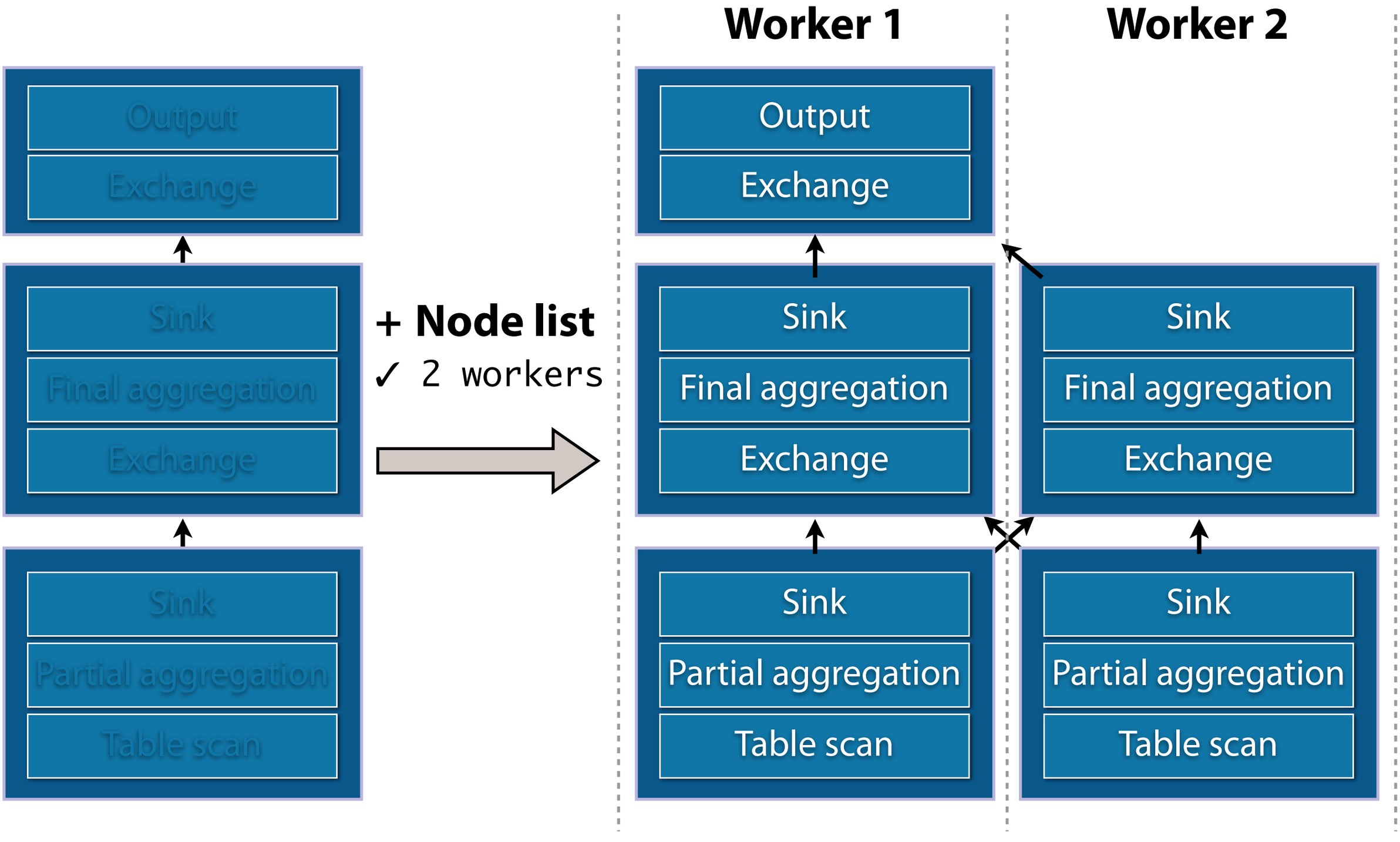
Stage-1

pipelined aggregation

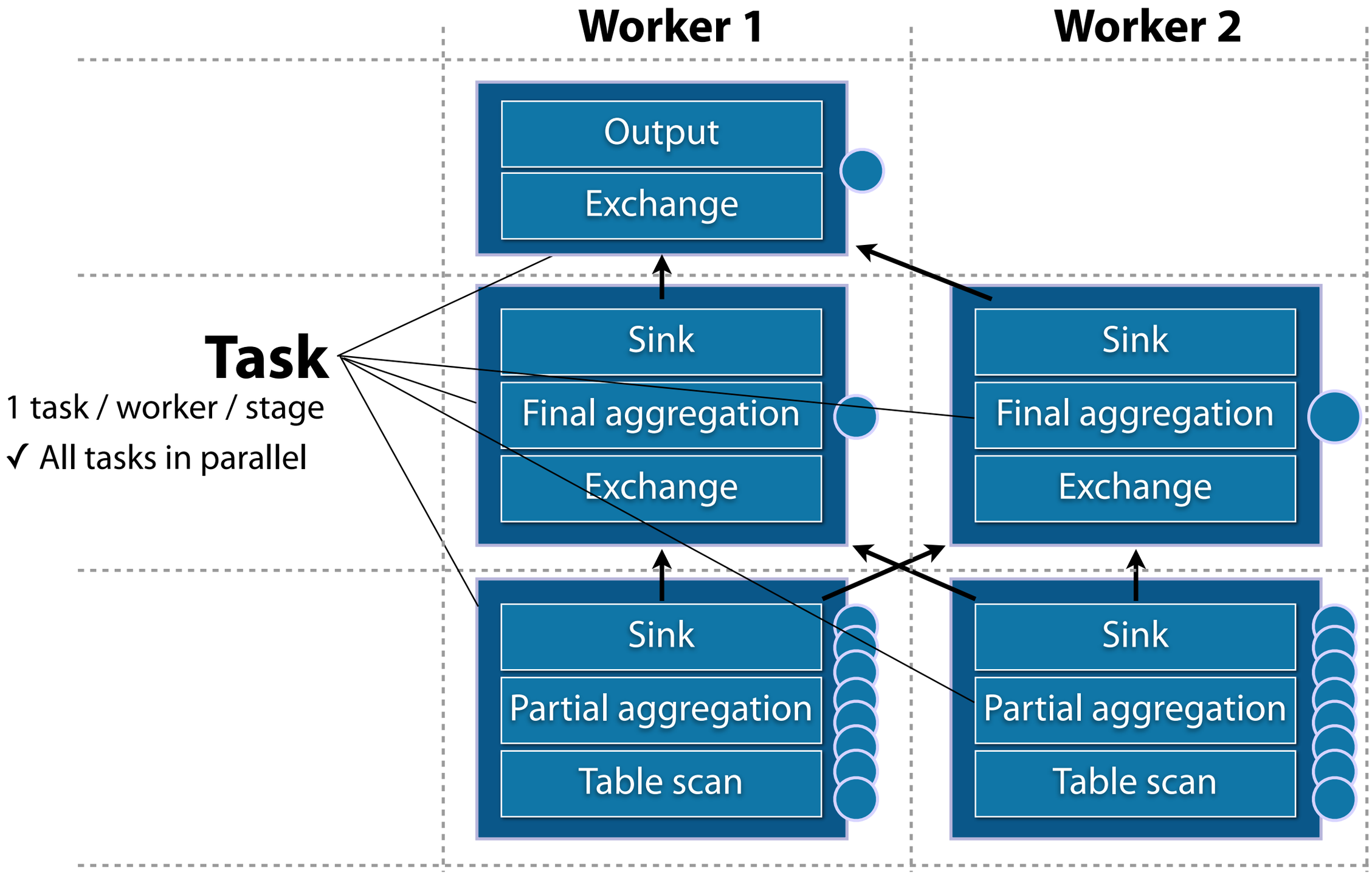
Stage-2

inter-worker data transfer

Execution Planner



Execution Planner - Tasks

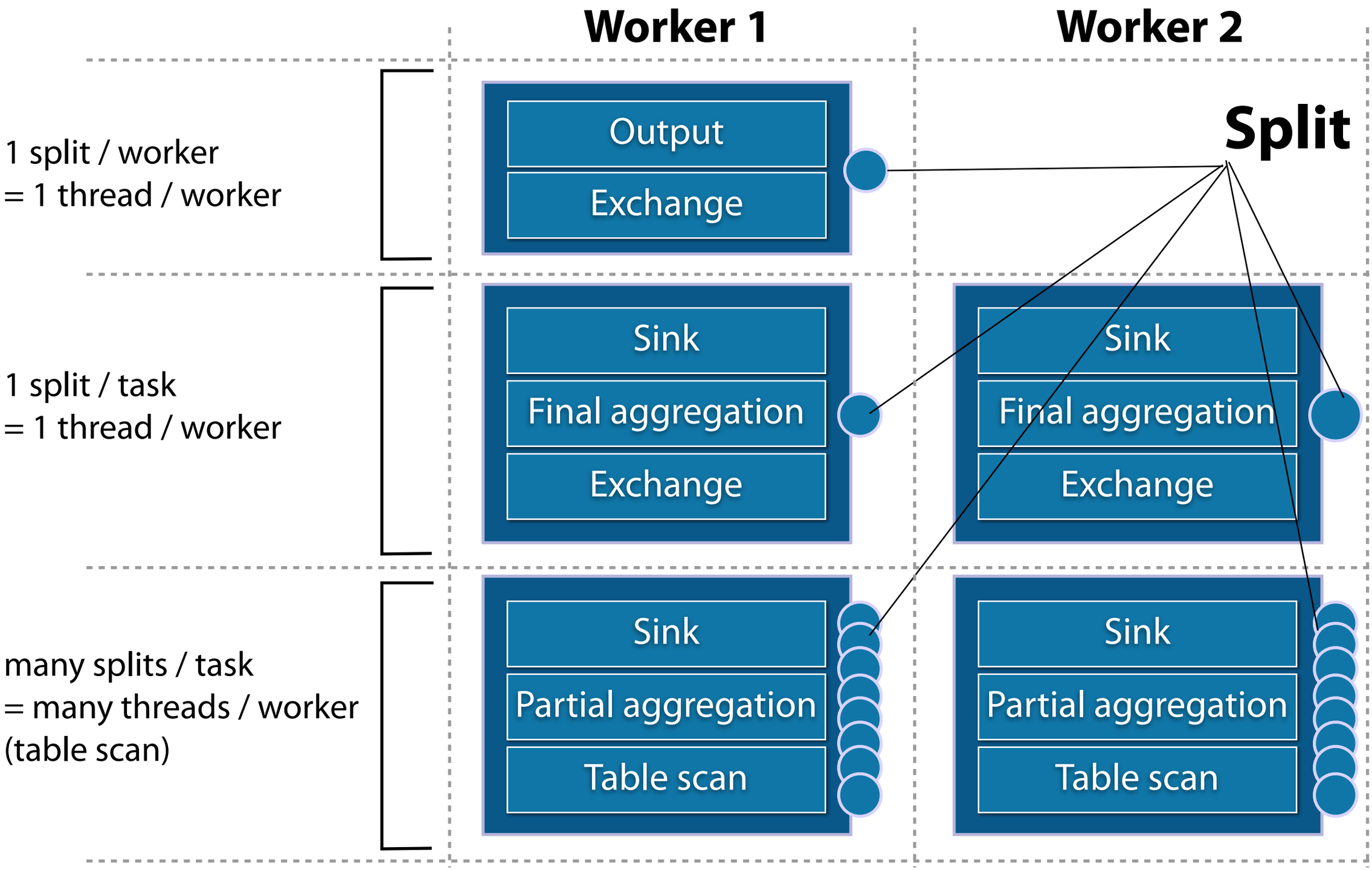


Task

1 task / worker / stage

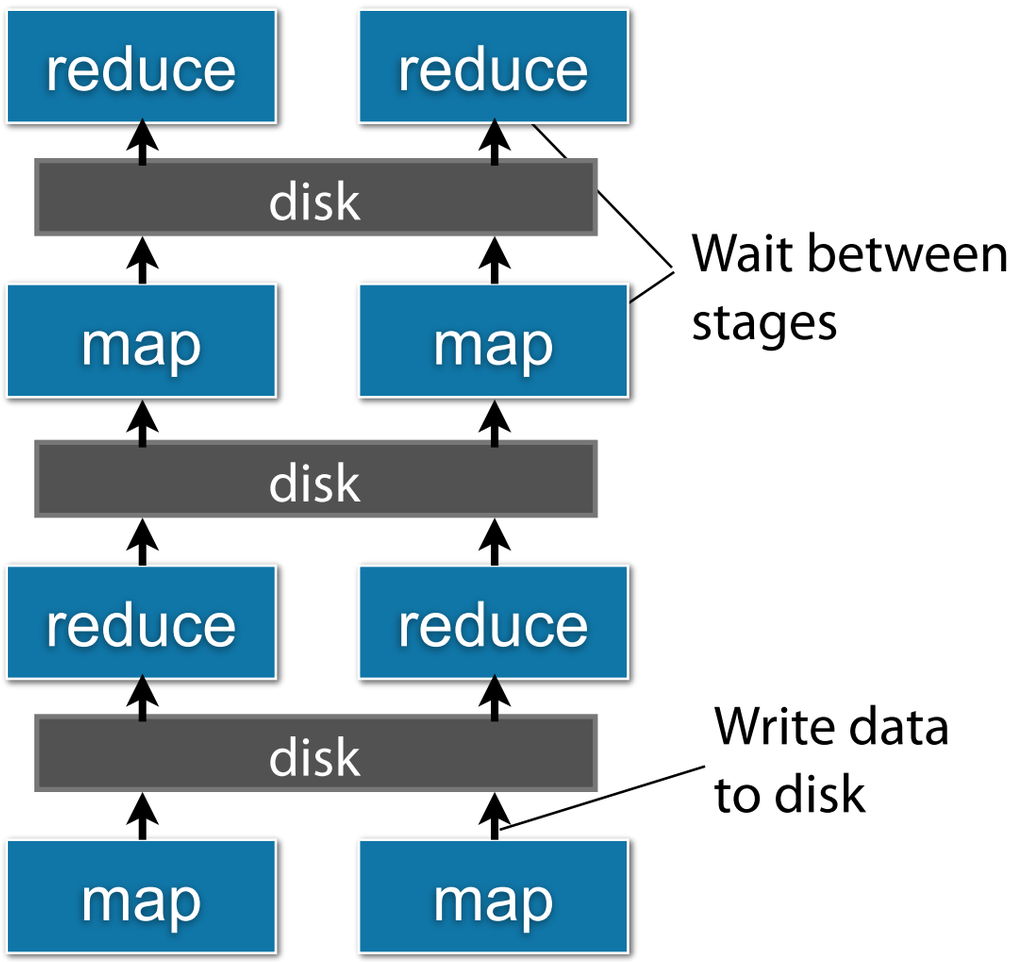
✓ All tasks in parallel

Execution Planner - Split

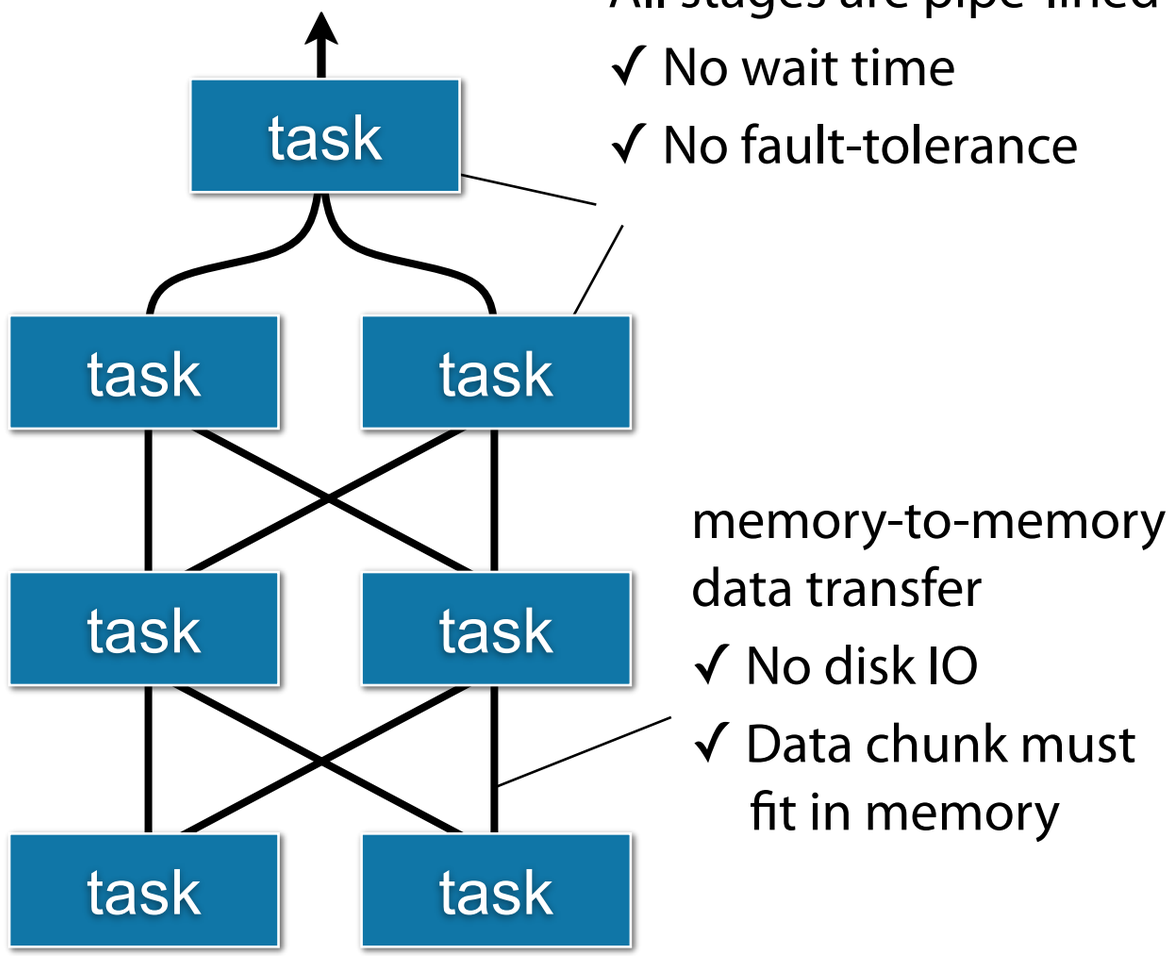


MapReduce vs. Presto

MapReduce



Presto



3. Query Execution

- **SQL is converted into stages, tasks and splits**
- **All tasks run in parallel**
  - No wait time between stages (pipelined)
  - If one task fails, all tasks fail at once (query fails)
- **Memory-to-memory data transfer**
  - No disk IO
  - If aggregated data doesn't fit in memory, query fails
    - Note: query dies but worker doesn't die. Memory consumption of all queries is fully managed

4. Monitoring & Configuration

Monitoring

- **Web UI**
  - basic query status check
- **JMX HTTP API**
  - GET /v1/jmx/mbean[/{objectName}]
    - com.facebook.presto.execution:name=TaskManager
    - com.facebook.presto.execution:name=QueryManager
    - com.facebook.presto.execution:name=NodeScheduler
- **Event notification (remote logging)**
  - POST http://remote.server/v2/event
    - query start, query complete, split complete

Configuration

- **Execution planning (for coordinator)**
  - query.initial-hash-partitions
    - max number of hash buckets (=tasks) of a GROUP BY (default: 8)
  - node-scheduler.min-candidates
    - max number of workers to run a stage in parallel (default: 10)
  - node-scheduler.include-coordinator
    - whether run tasks only on workers or include coordinator
  - query.schedule-split-batch-size
    - number of splits of a stage to start at once

Configuration

- **Task execution (for workers)**
  - task.cpu-timer-enabled
    - enable detailed statistics (causes some overhead) (default: true)
  - task.max-memory
    - memory limit of a task especially for hash tables used by GROUP BY and JOIN operations (default: 256MB)
    - enlarge if you get “Task exceeded max memory size” error
  - task.shard.max-threads
    - max number of threads of a worker to run active splits (default: number of CPU cores * 4)

5. Roadmap

A report of Presto Meetup 2014

"Presto, Past, Present, and Future" by Dain Sundstrom at Facebook

http://www.slideshare.net/dain1/presto-meetup-20140514-34731104

Presto's future

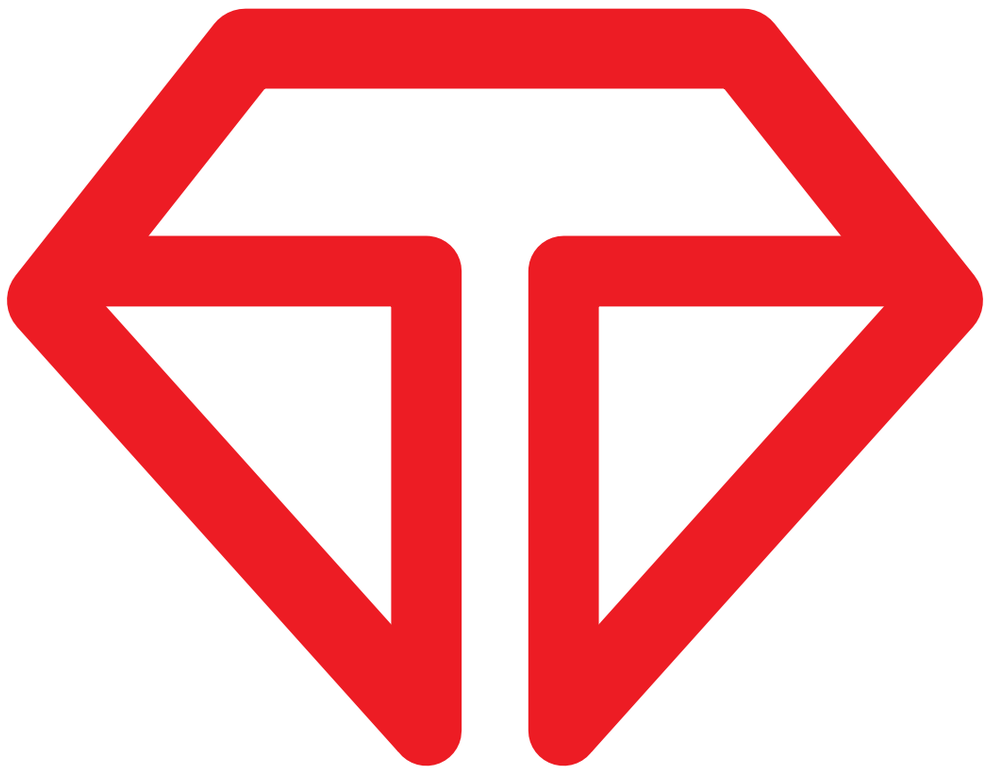
- **Huge JOIN and GROUP BY**
  - Spill to disk
- **Task recovery**
- **CREATE VIEW (※implemented)**
- **Native store (※implemented)**
  - Fast data store in Presto workers
  - to cache hot data
- **Authentication and permissions**

Presto's future

- **DDL/DML statements**
  - CREATE TABLE with partitioning
  - DELETE and INSERT
- **Plugin repository**
- **CLI plugin manager**
- **JOIN and aggregation pushdown**
- **Custom optimizers**

Links

- **Web site & document**
  - http://prestodb.io
- **Mailing list**
  - https://groups.google.com/group/presto-users
- **Github**
  - https://github.com/facebook/presto
- **Guidelines for contribution**
  - https://github.com/facebook/presto/blob/master/CONTRIBUTING.md



TREASURE

Cloud service for the entire data pipeline,
including Presto. We're hiring!

Check: www.treasuredata.com