
**Università degli Studi di Milano**

Facoltà di Scienze e Tecnologie

Corso di Laurea in Sicurezza dei sistemi e delle reti informatiche


# Apache Hive and Apache Druid performance testing for MIND Foods HUB Data Lake


**Supervisor**
Prof. Paolo Ceravolo

**Co-supervisor**
Filippo Berto

**Graduand**
Gabriele D'Arrigo
909953

# MIND FOODS HUB

An international, interdisciplinary project that operates in the context of the Milan Innovation District with the goal of "implementing a **computational infrastructure** to model, engineer and distribute data about plant phenotyping".

arexpo

UNIVERSITÀ DEGLI STUDI DI MILANO

TIM

ISTITUTO AUXOLOGICO ITALIANO
Istituto di ricovero e cura a carattere scientifico

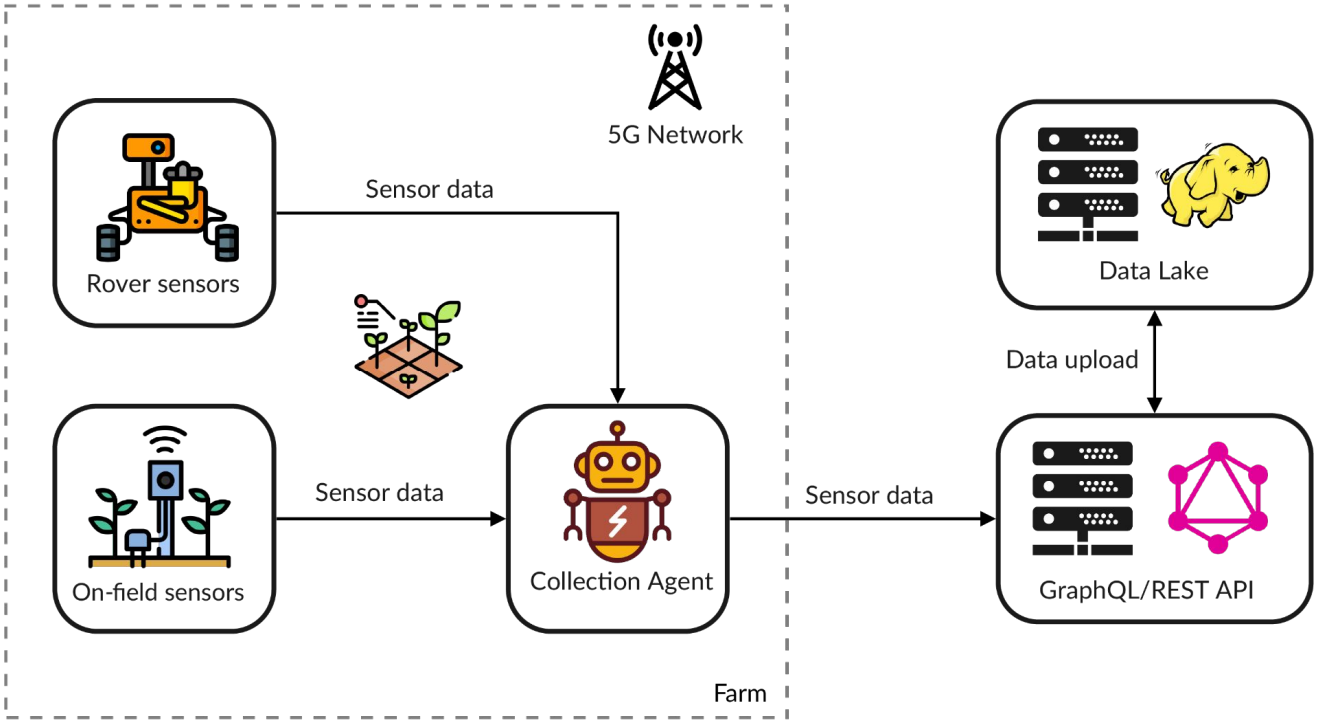
indena
SCIENCE IS OUR NATURE. SINCE 1921
100 YEARS

MOLINO FILIPPINI

TECNOALIMENTI

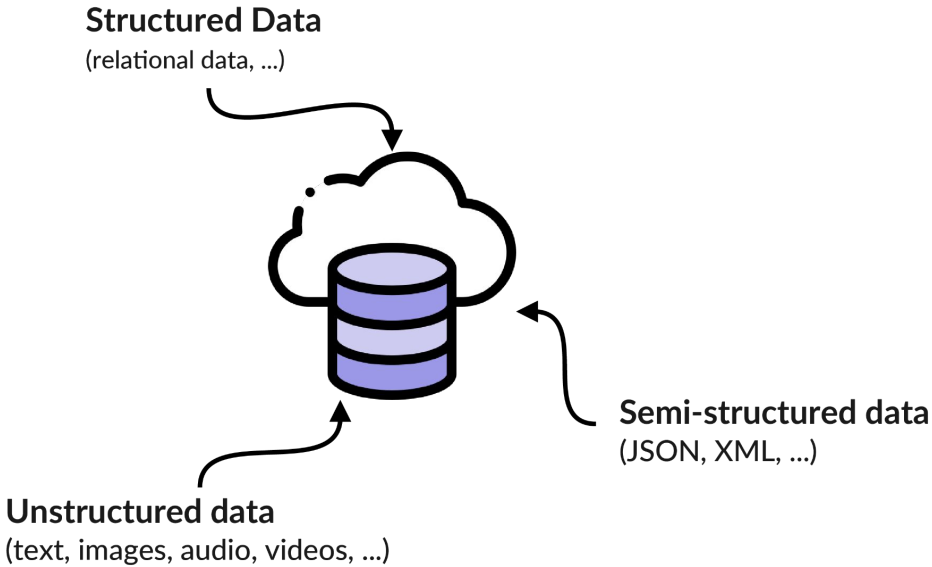
AGRICOLA MODERNA

# MFH computing infrastructure



# Data Lake

In Big Data, a Data Lake is a repository that stores **large** quantities and varieties of data in their **raw** format, independently from their source or structure.



# Apache Hive


A data warehouse software that facilitates reading, writing, and managing **large** datasets residing in **distributed** storage using **SQL**.

**Use cases:** ETL tasks, reporting, and data analysis in batch mode with SQL.

**Data format:** CSV/TSV, JSON, Apache Parquet, Apache ORC, and others.

**Data model**: databases, tables, views, partitions and buckets.

**Storage:** distributed storages like Hadoop HDFS.

**Data ingestion:** batch mode with MapReduce.

# Research Goals

**Problem**: Apache Hive is difficult to maintain and slow on simple aggregation queries.

**Research**: find, implement and test an alternative platform that satisfies these requirements:

1. **Maintainability**: the platform must be easy to configure and deploy on the MIND Foods HUB Hadoop cluster
2. **Performance**: the platform should provide sub-second aggregations queries

# Apache Druid

A real-time database to power modern analytics applications.

**Use cases:** Real time analysis, backend for highly concurrent APIs, low latency queries

**Data format:** CSV/TSV, JSON, Apache Parquet, Apache ORC, Protobuf, and others.

**Data model**: datasources, with time-based partitioning

**Storage:** distributed storage like Hadoop HDFS or Amazon S3

**Data ingestion:** real time ingestion in streaming mode, or batch ingestion

# Performance testing

In software, performance testing is a type of non-functional testing that measures a system's behaviour under satisfactory and unsatisfactory conditions.

1. Gather testing **requirements**
2. Create a **benchmark** representative of how the system is used in the field
3. Measure the system's performance by collecting various time-related metrics, like **response time**, throughput, and concurrency

# Apache Hive and Apache Druid performance testing

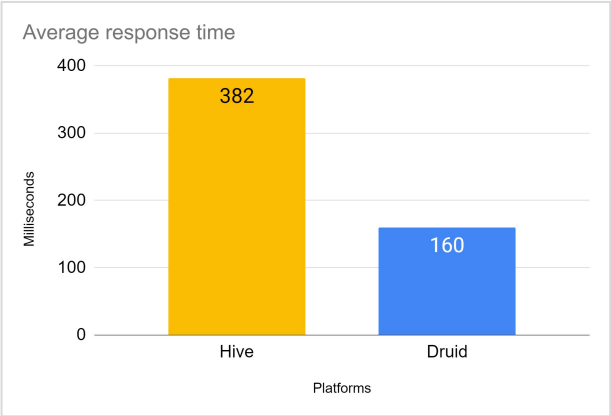
1. Provision with Docker of **Hadoop**, **Hive** and **Druid** on the SESAR Lab cluster.
2. Generate synthetic data:

   I wrote a Node.js application to generate a dataset of **50 million rows**.
3. Ingest data with specific schema **optimization.**
4. Prepare the **test queries.**
5. Run the **HTTP performance testing** using Apache JMeter:

   I wrote a Node.js application to execute HIVE SQL statements via HTTP.

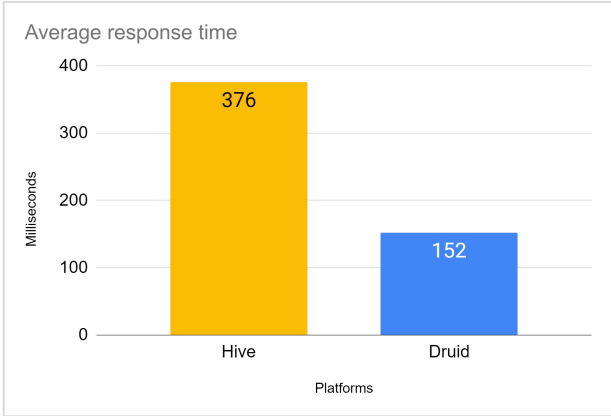
# Results - 1

Query 1



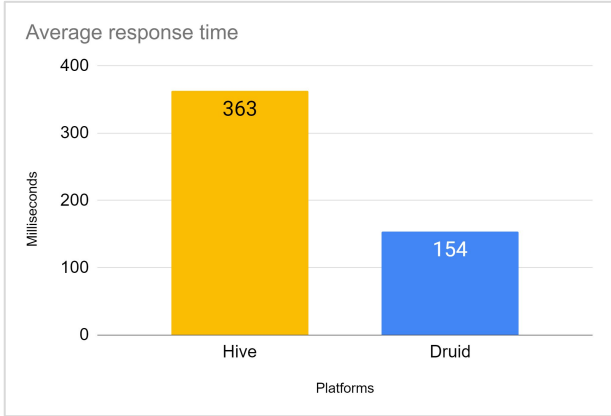

```
SELECT *
FROM mfh.dl_measurements
WHERE insertion_date >= '2021-12-01'
AND insertion_date <= '2021-12-31'
AND double_value IS NOT NULL
LIMIT 100;
```

Query 2




```
SELECT *
FROM mfh.dl_measurements
WHERE insertion_date >= '2021-12-01'
AND insertion_date <= '2021-12-02'
AND str_value IS NOT NULL
AND start_timestamp IS NULL
LIMIT 100;
```

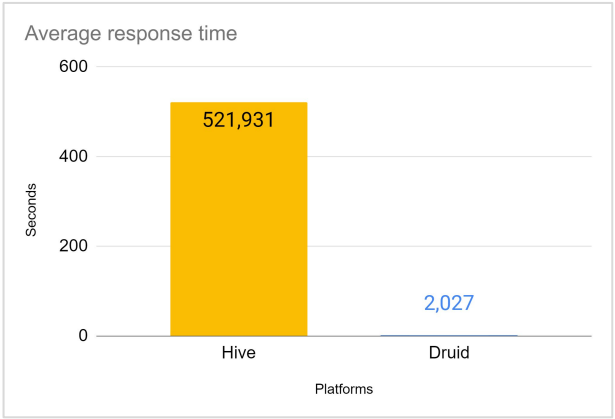
Query 3




```
SELECT *
FROM mfh.dl_measurements
WHERE insertion_date >= '2021-12-01'
AND insertion_date <= '2021-12-02'
AND str_value IS NOT NULL
AND start_timestamp IS NOT NULL
LIMIT 100;
```

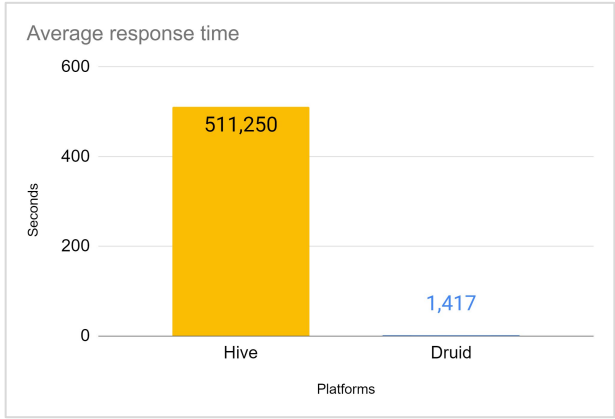
# Results - 2

## Query 4



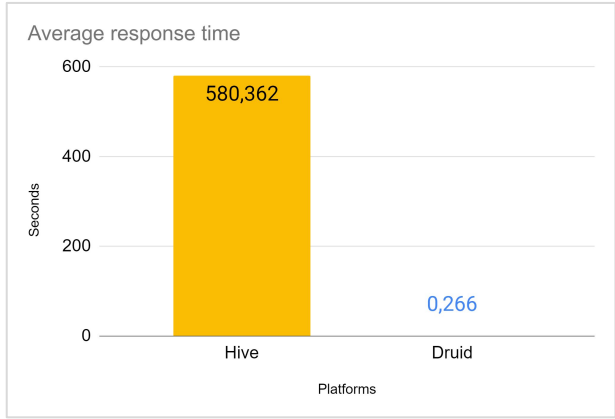

```
SELECT location_id, location_name,
location_botanic_name,
location_cultivation_name, COUNT(*) AS
number_of_measurements
FROM dl_measurements
WHERE location_id = 'cassoni_sx'
GROUP BY location_id, location_name,
location_botanic_name,
location_cultivation_name;
```

## Query 5




```
SELECT sensor_id, sensor_type,
sensor_desc_name, COUNT(*) AS
number_of_measurements
FROM dl_measurements
GROUP BY sensor_id, sensor_type,
sensor_desc_name;
```

## Query 6




```
SELECT sensor_id, location_cultivation_name,
AVG(double_value) AS average
FROM dl_measurements
WHERE sensor_id =
'TS_0310B473-depth_soiltemperature'
AND location_id = 'cassoni_sx'
AND location_cultivation_name = 'Rubiaceae'
GROUP BY sensor_id, location_id,
location_cultivation_name;
```

# Conclusions

Apache Druid achieved an indisputable performance increment over Hive:

**Time queries:** from an average response time of **372** ms to **155** ms

**Aggregate queries:** from an average response time of **9** m to **1,23** s

Also, Apache Druid's maintainability is better, thanks to its modern architecture, the official support for Docker, and its detailed and comprehensive documentation.

# Thank you! 

## Questions?