




# **BATCH PROCESSING**

**Presentato da : Dissan Uddin Ahmed**

**Università Tor Vergata | 2024**



# OVERVIEW

- Introduzione
  - Obiettivi
  - Dataset
  - Deployment
  - Architettura
  - Ingestion
  - Processamento
  - Storage
  - Risultati
- 

# INTRODUZIONE

**Gli hard disk, sono i componenti che vengono sostituiti con maggiore frequenza. Sono la causa della maggior parte dei fallimenti nei datacenter.**



# OBIETTIVI

**Rispondere a 3 query usando il framework spark per il batch processing caricando i risultati su HDFS**

## ● Query I

Per ogni giorno e per ogni vault si vuole determinare i vault che hanno avuto dai 2 ai 4 fallimenti

## ● Query II

Si vuole fare la top 10 dei 50 modelli di hard disk che hanno avuto più fallimenti. E la top 10 dei vault con relativa lista dei modelli di hard disk

## ● Query III

Computare il 25, 50, 75 percentile del numero di ore in funzione per gli hard disk che hanno subito un fallimento e per gli hardisk che non lo hanno subito

# DATASET

## Disk failures Blackbaze

Il dataset contiene i dati di 23 giorni di funzionamento degli hard disk in un datacenter, in questo progetto verranno considerati: date, serial\_number, model, failure, vault\_id e s9\_power\_on\_hours

# DEPLOYMENT

## Docker Compose

Il sistema usa un ambiente containerizzato, viene usato docker compose per coordinare i nodi presenti nel sistema

Hadoop-cluster

Spark-cluster

Apache/nifi

Cassandra



# ARCHITETTURA



## Apache nifi

Per il preprocessing e ingestion



## Apache Hadoop

File system distribuito per rendere persistente i risultati



## Apache Spark

Framework per il processing dei dati

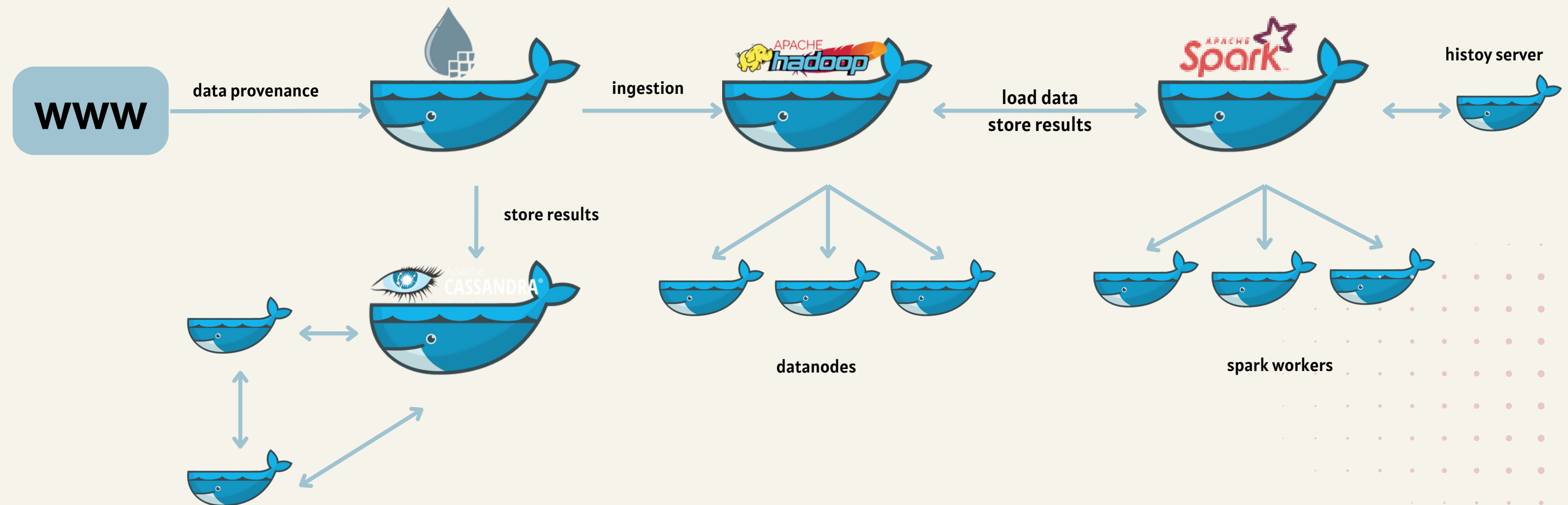


CASSANDRA

## Apache Cassandra

Datastore noSql usato per backup dei dati

# ARCHITETTURA











# INGESTION







## Apache Nifi









Frame work usato per prendere i dati dal web e fare preprocessing per alleggerire il carico di lavoro al layer di processamento

Receive Raw Multi-Modal Data			
<div><div> 0</div><div> 0</div><div> 12</div><div> 0</div><div> 0</div><div> 0</div></div>			
Queued	5,040,385 (400 GB)		
In	0 (0 bytes) → 0		5 min
Read/Write	13 GB / 10 GB		5 min
Out	2 → 1,523 (102 GB)		5 min
No comments specified			

Name	Video
Queued	15 (301 GB) → 1

Extract Semantic Meaning and Index			
 0  0  32  0  0  0			
Queued	13 (80 GB)		
In	117 (540 GB) → 1	5 min	
Read/Write	502 GB / 12 GB	5 min	
Out	0 → 0 (0 bytes)	5 min	
No comments specified			

Name	Unstructured Text
Queued	52 (12 MB) → 1

Chunk, Vectorize, Store Embeddings			
 0  0  18  0  0  0			
Queued	80 (23 MB)		
In	100 (54 MB) → 1	5 min	
Read/Write	3 GB / 2 GB	5 min	
Out	0 → 0 (0 bytes)	5 min	
No comments specified			

## INPUT-OUT



## INGESTION

## GET DATASET

	<b>InvokeHTTP</b> InvokeHTTP 1.26.0 org.apache.nifi - nifi-standard-nar
In	0 (0 bytes) 5 min
Read/Write	0 bytes / 0 bytes 5 min
Out	0 (0 bytes) 5 min
Tasks/Time	0 / 00:00:00.000 5 min

Name Original, Response  
Queued 0 (0 bytes)

	<b>CompressContent</b> CompressContent 1.26.0 org.apache.nifi - nifi-standard-nar
In	0 (0 bytes) 5 min
Read/Write	0 bytes / 0 bytes 5 min
Out	0 (0 bytes) 5 min
Tasks/Time	0 / 00:00:00.000 5 min

Name success  
Queued 0 (0 bytes)

	<b>UnpackContent</b> UnpackContent 1.26.0 org.apache.nifi - nifi-standard-nar
In	0 (0 bytes) 5 min
Read/Write	0 bytes / 0 bytes 5 min
Out	0 (0 bytes) 5 min
Tasks/Time	0 / 00:00:00.000 5 min

## PROCESSING GROUP

Ingestion-flow	
Queued	0 (0 bytes)
In	0 (0 bytes) → 1 5 min
Read/Write	0 bytes / 0 bytes 5 min
Out	2 → 0 (0 bytes) 5 min

From Conver to csv  
Queued 0 (0 bytes)

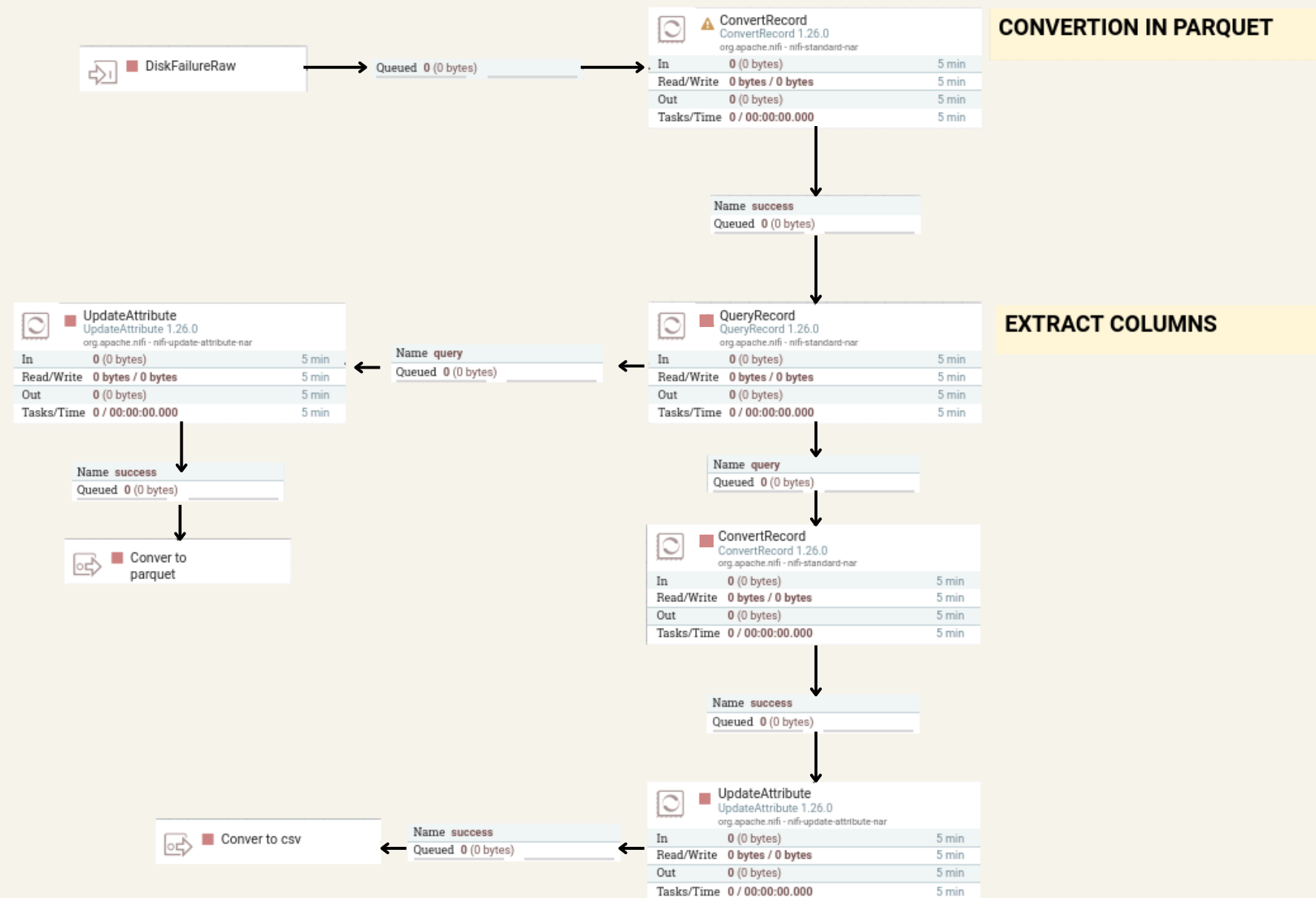
	<b>PutHDFS</b> PutHDFS 1.26.0 org.apache.nifi - nifi-hadoop-nar
In	0 (0 bytes) 5 min
Read/Write	0 bytes / 0 bytes 5 min
Out	0 (0 bytes) 5 min
Tasks/Time	0 / 00:00:00.000 5 min

## STORE PREPROCESSED DATA

	<b>PutHDFS</b> PutHDFS 1.26.0 org.apache.nifi - nifi-hadoop-nar
In	0 (0 bytes) 5 min
Read/Write	0 bytes / 0 bytes 5 min
Out	0 (0 bytes) 5 min
Tasks/Time	0 / 00:00:00.000 5 min

# INGESTION

Preprocessing  
nifi



# PROCESSAMENTO

## Apache Spark



Frame work usato per processare le query del progetto. Sono state usate le api dataframe per le ottimizzazioni interne fatte dal framework. Query I e II sono disponibili anche in SQL.

Query I

Query II

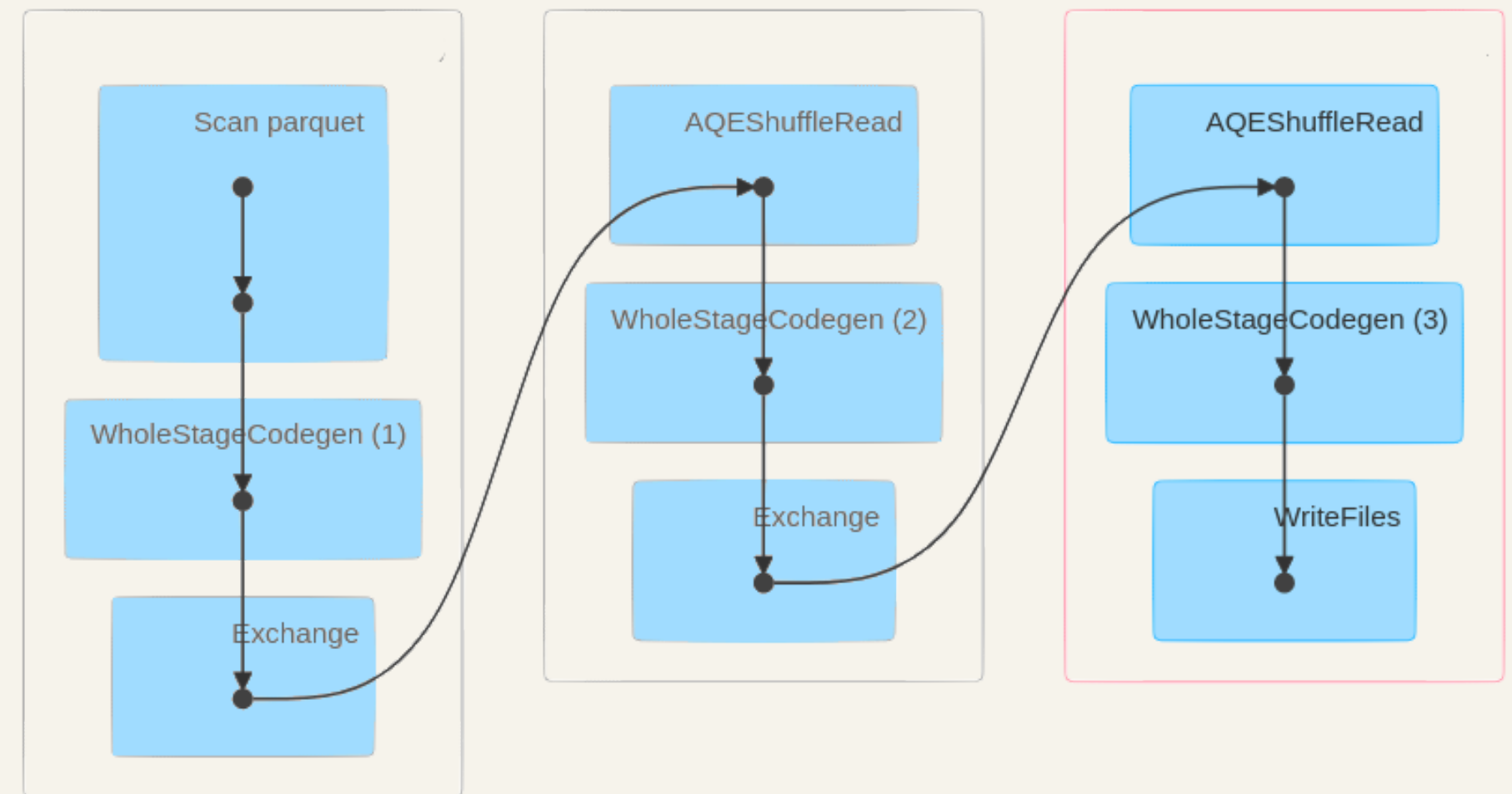
Query III

# PROCESSAMENTO

## Query 1



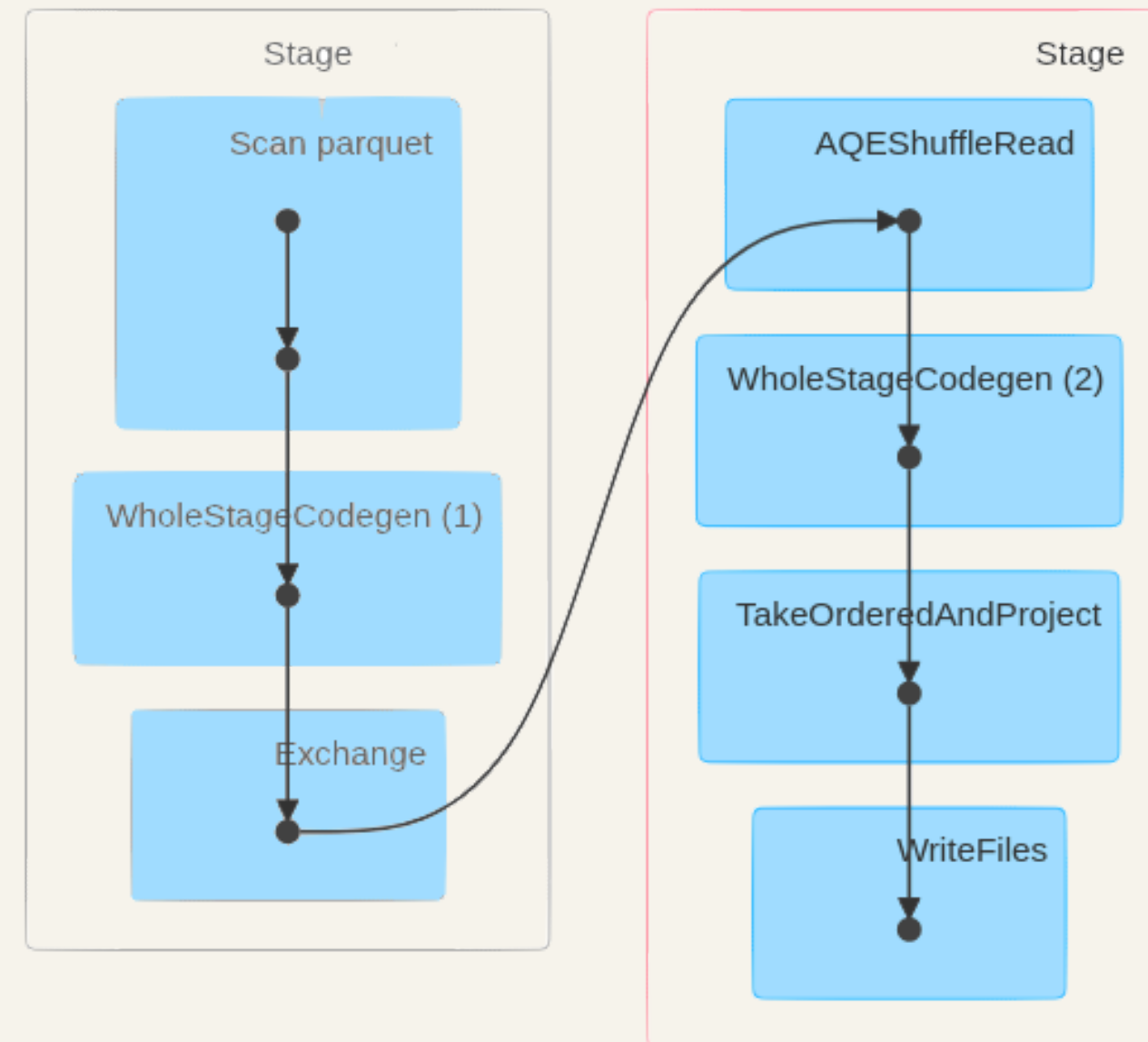
Per ogni giorno e per ogni vault  
determinare la lista dei vault che  
hanno subito esattamente 2,3,4  
fallimenti



# PROCESSAMENTO

## Query II prima parte

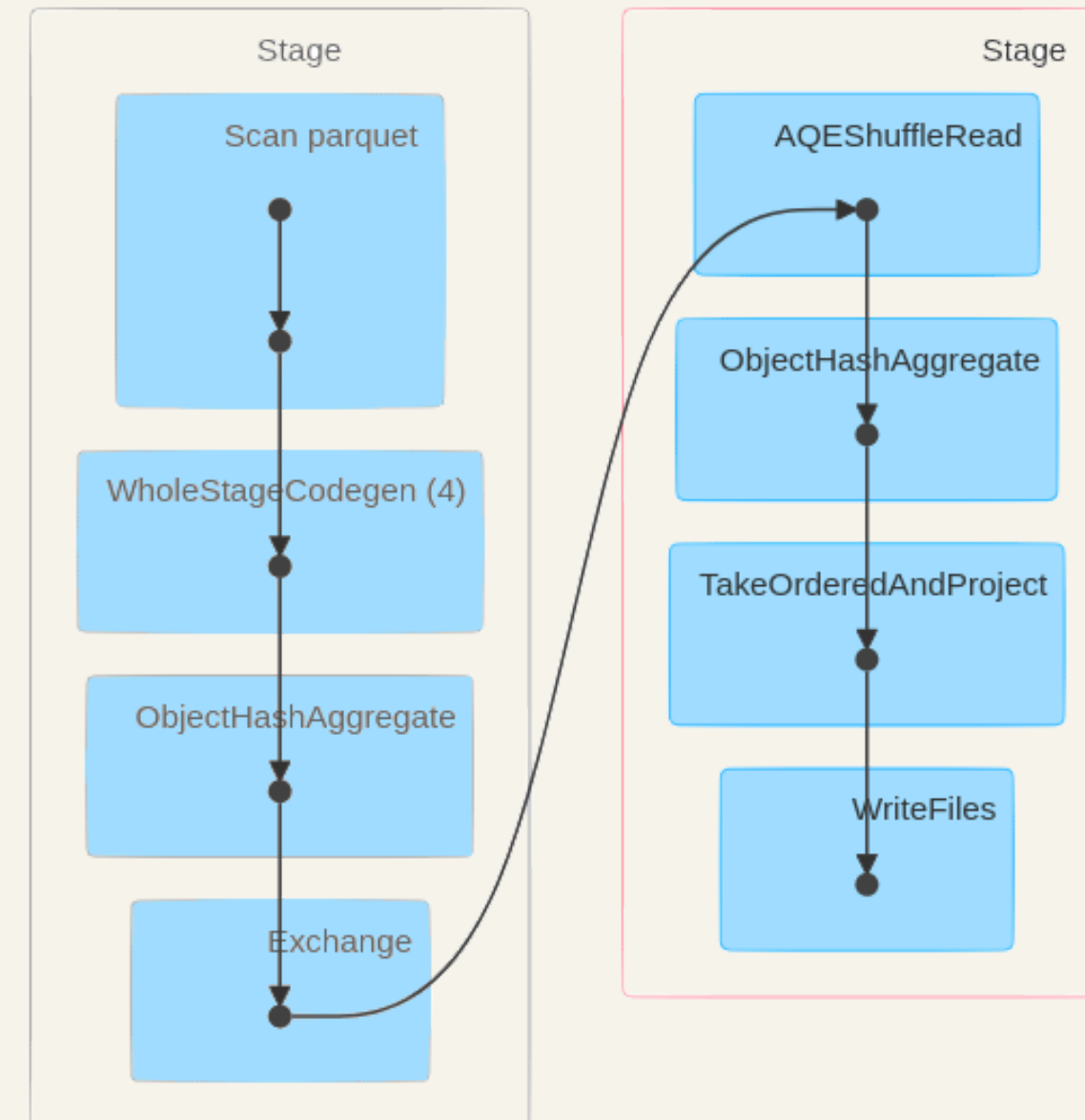
Determinare la lista dei modelli che hanno subito più fallimenti



# PROCESSAMENTO

## Query II seconda parte

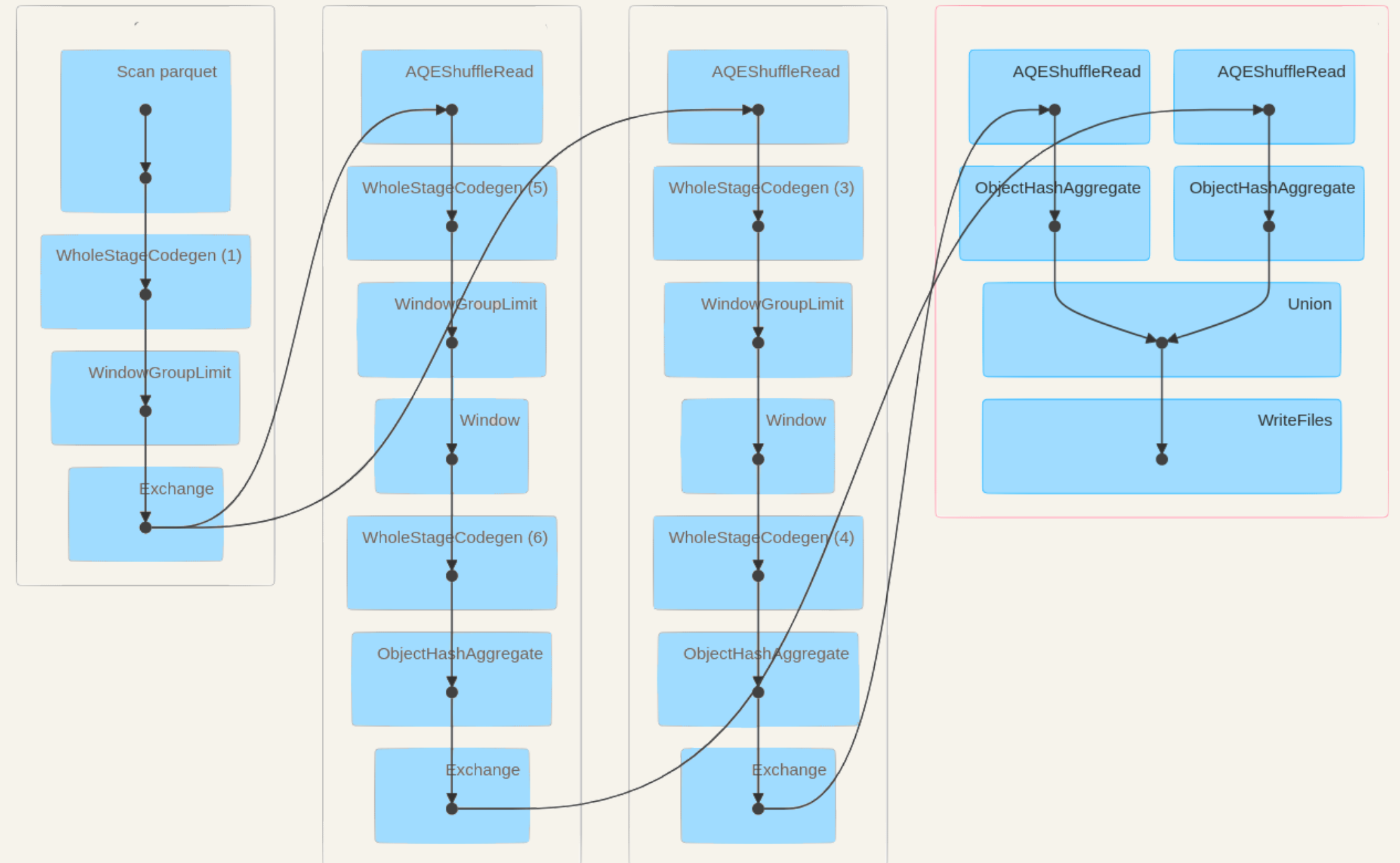
Determinare la lista dei top 10 vault che hanno subito più fallimenti con relativa lista univoca dei modelli



# PROCESSAMENTO

## Query III

Calcolare i min, max 25,50,75 percentili dei dischi che hanno subito fallimento e che non hanno subito fallimento

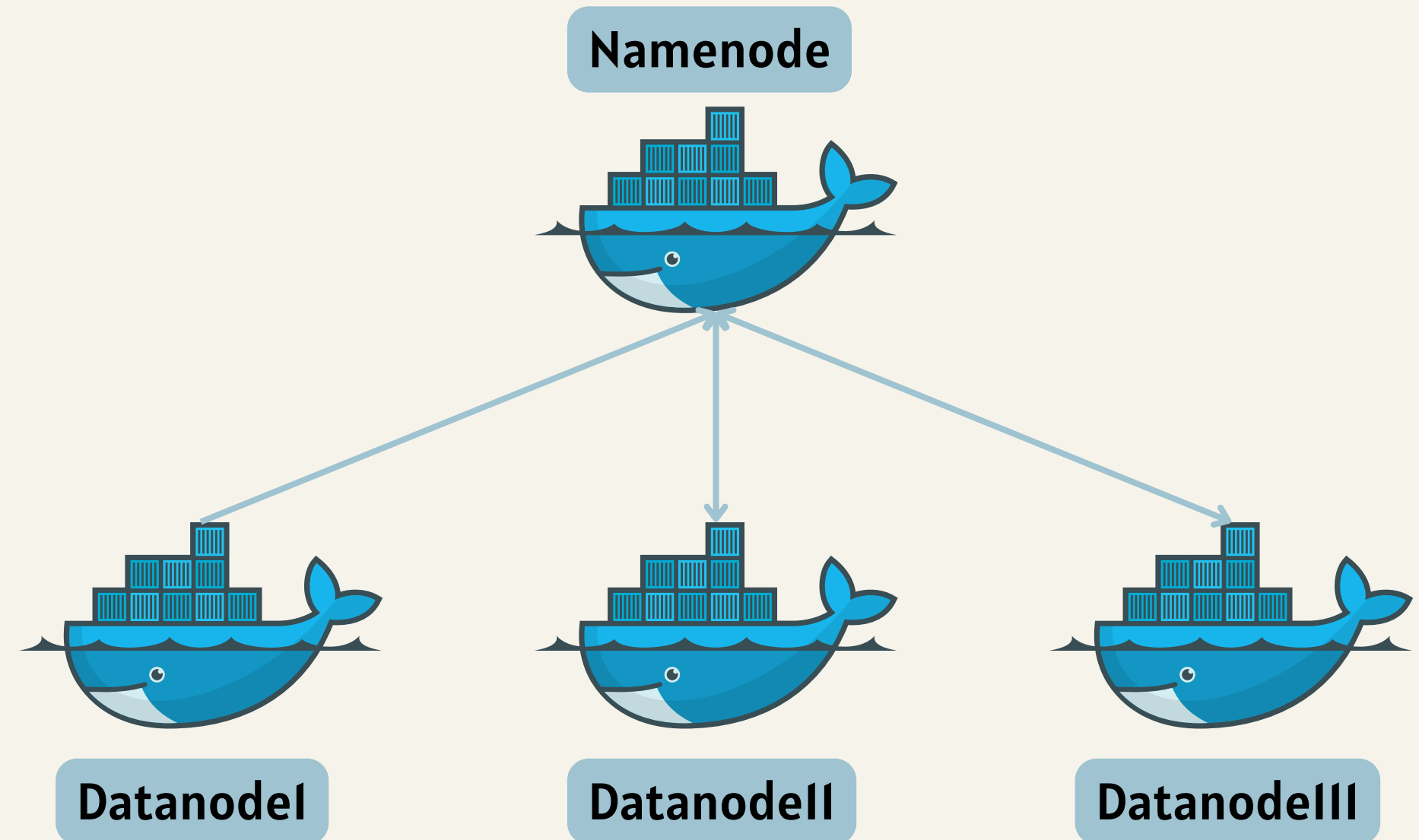




# STORAGE

## HDFS

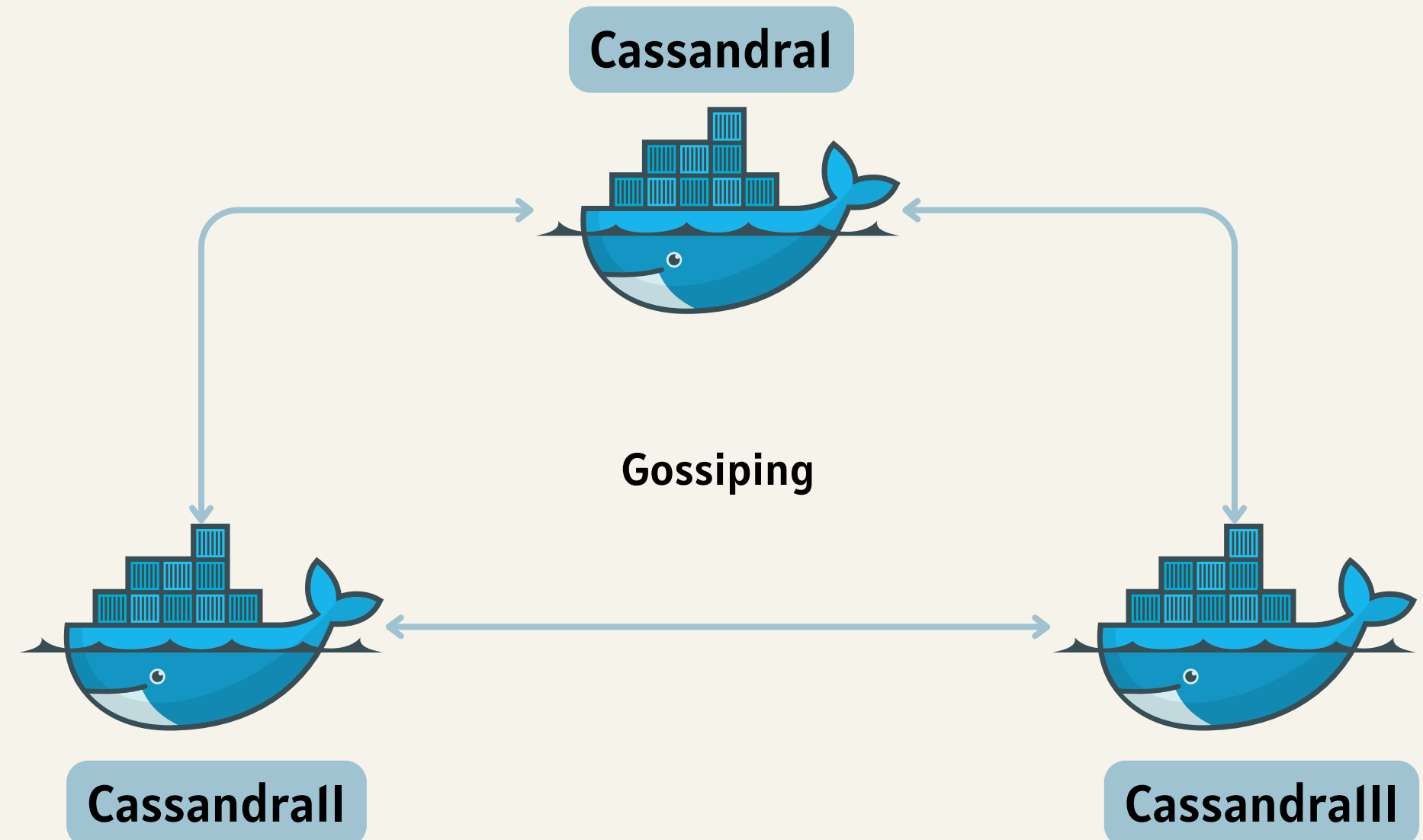
HDFS è stato usato per salvare le query in un filesystem distribuito



# STORAGE

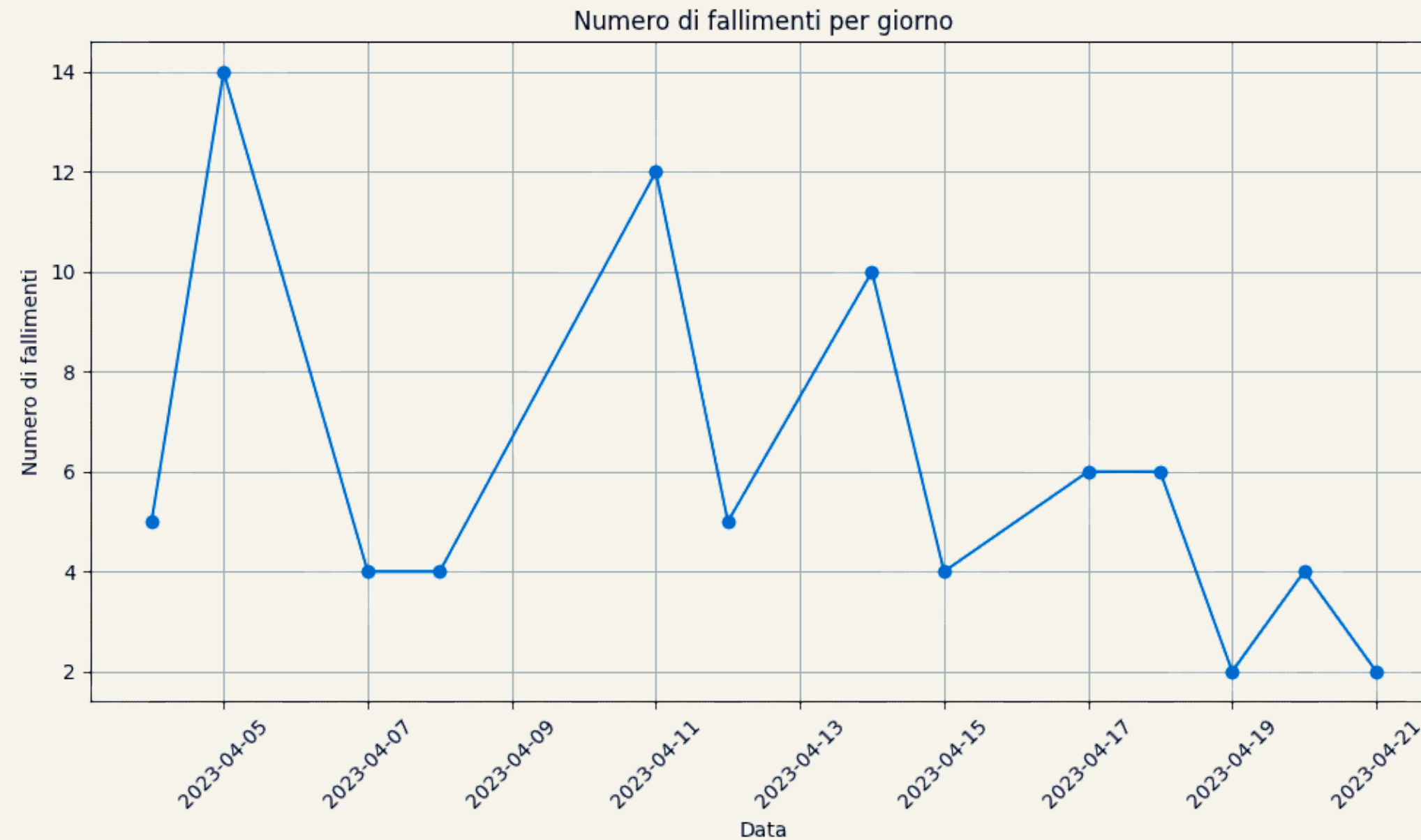
## Cassandra

Apache cassandra datastore noSql che offre il linguaggio cql che ha una sintassi simile a SQL. Rendendo il sistema maggiormente accessibile a chi ha familiarità con SQL



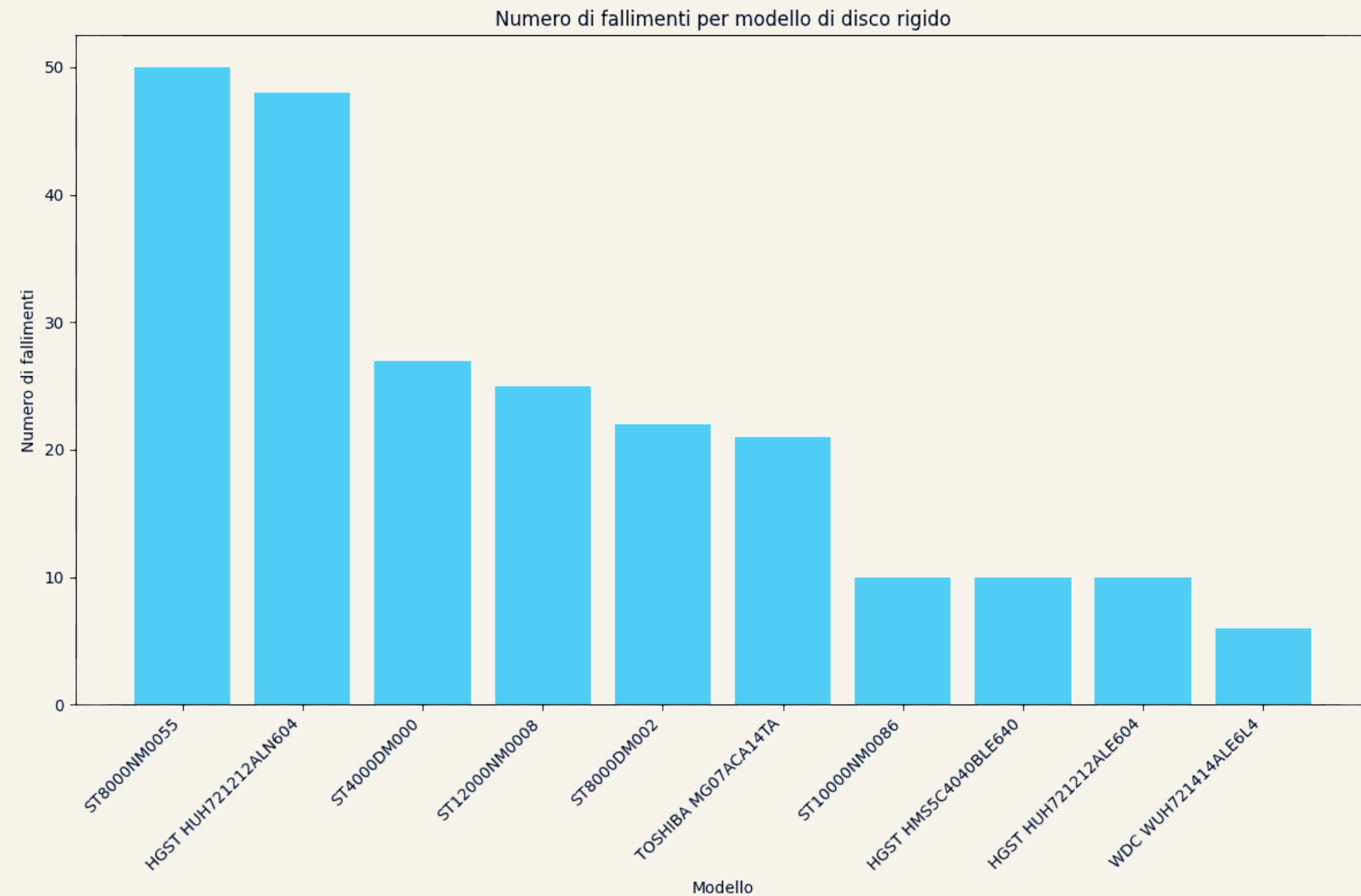
# RISULTATI

## Query I



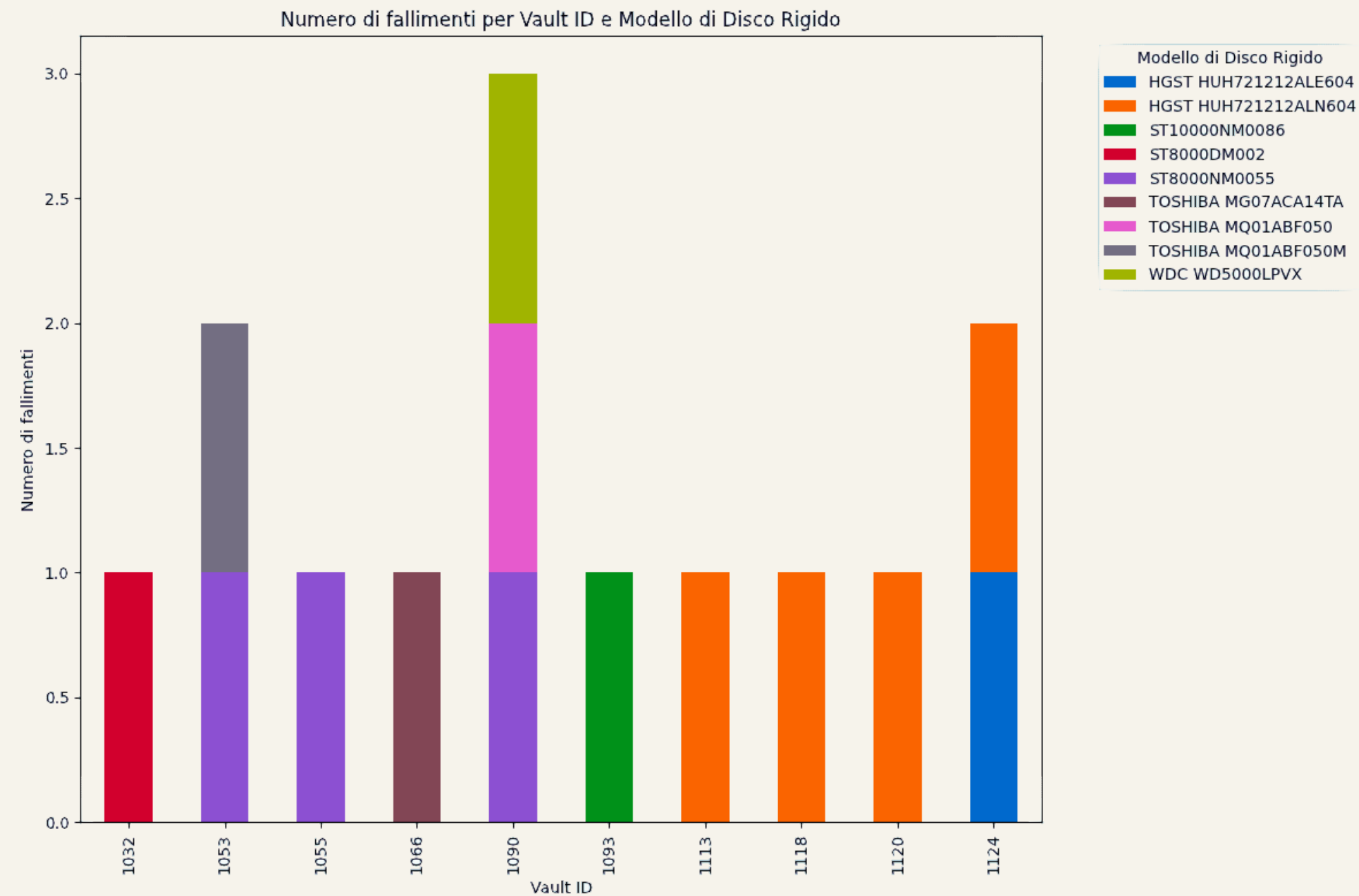
# RISULTATI

Query II  
top 10 modelli



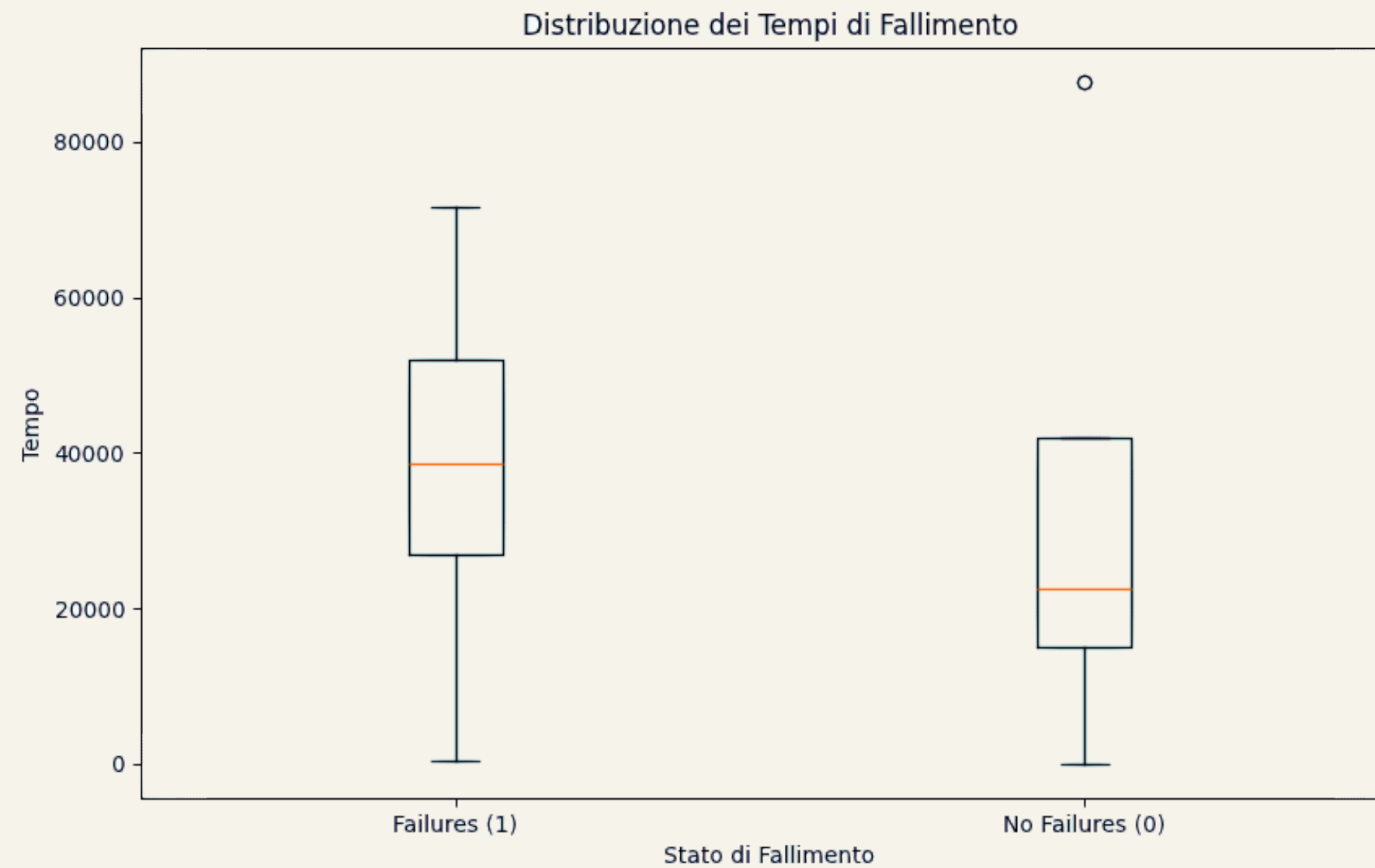
# RISULTATI

Query II  
top 10 vault



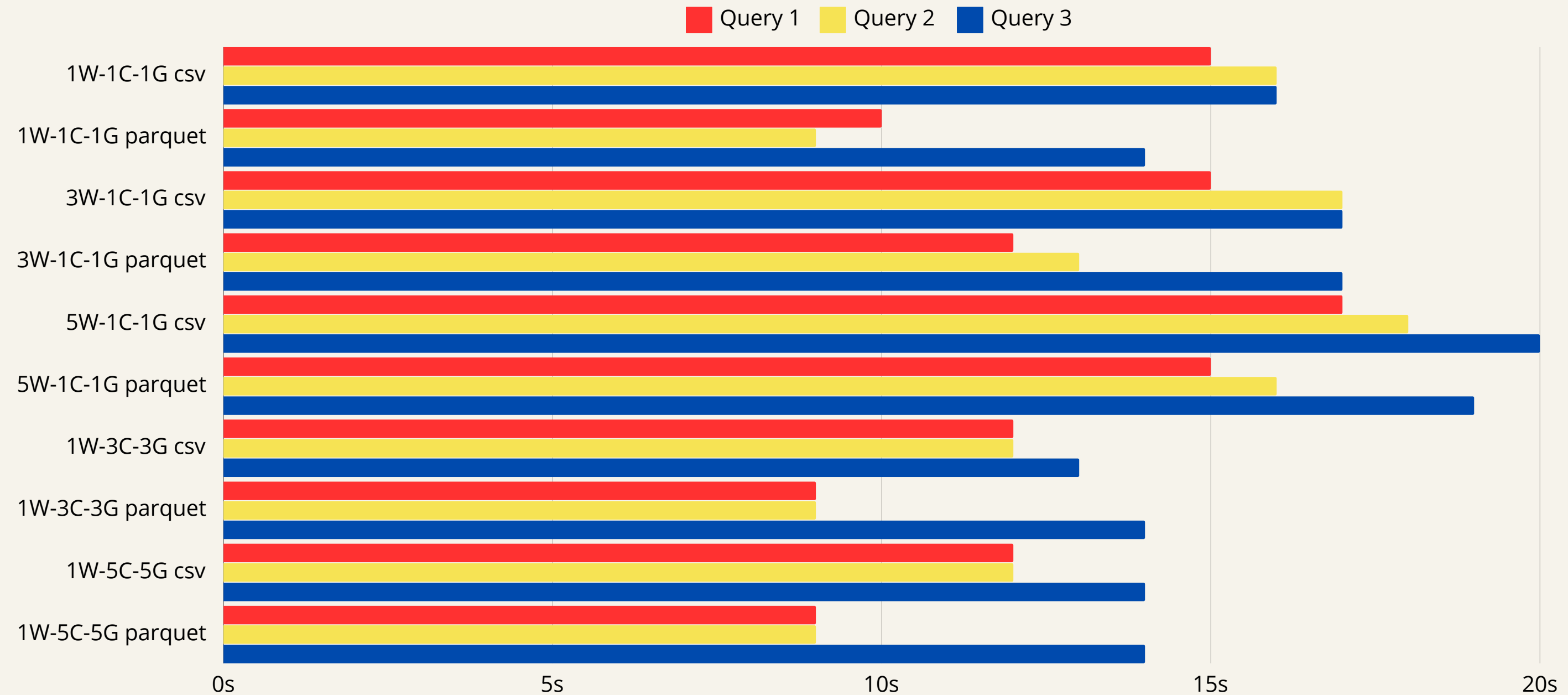
# RISULTATI

## Query III



# RISULTATI

## PERFORMANCE



The background features three vertical stripes on the left: a wide pink stripe, a narrower blue stripe, and a medium-width beige stripe. The right side of the slide is white, with a pattern of small pink dots arranged in a grid that fades out towards the right edge.

**Università Tor Vergata | 2024**

**GRAZIE**

**Presentato da : Dissan Uddin Ahmed**