REAL-TIME ANALYSIS OF DISK MONITORING EVENTS WITH APACHE FLINK

Emanuele Valzano - Matricola 0341634

AGENDA

TOPICS COVERED

INTRODUCTION

REAL-TIME SIMULATION

SYSTEM ARCHITECTURE

PREPROCESSING

PROCESSING

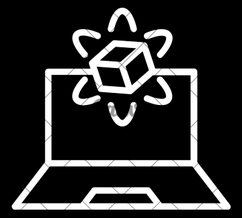
METRICS EVALUATION

INTRODUCTION

The goal of the project is to answer specific queries related to real-time telemetry data events from approximately 200,000 hard disks in data centers managed by Backblaze.

Events are generated in real time and are processed leveraging Apache Flink as the Stream Processing Framework.

REAL-TIME SIMULATION



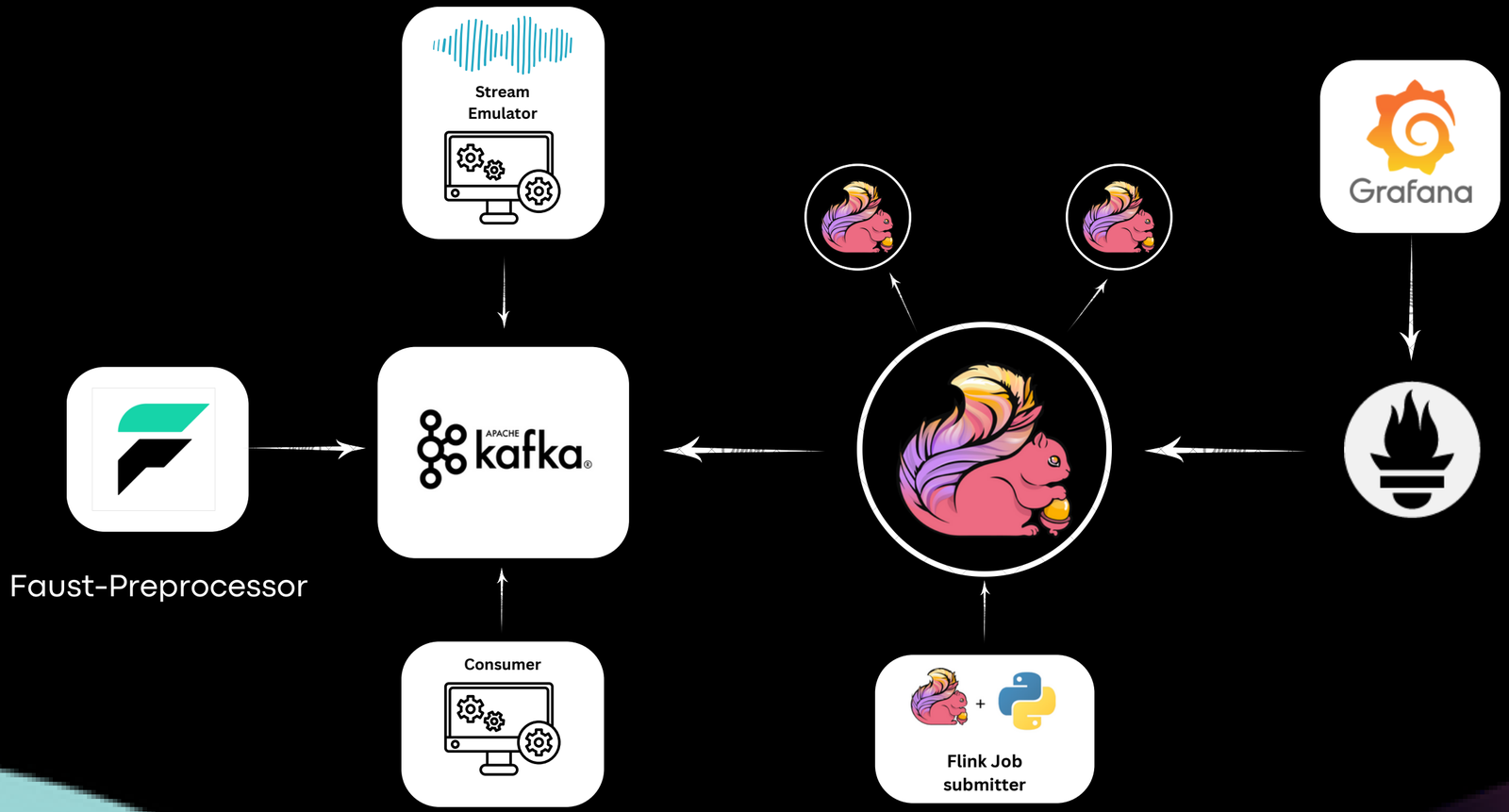
A stream emulator reads the dataset from a local source
and iterates over the rows to produce events in real-time to a **Kafka Topic**

Two modes of execution:

- **Fast mode**: tuples within the same day are produced in a burst.
- **Non-Fast mode**: tuples are interleaved based on a random uniform distribution.

Flushin intervals to avoid the producer queue to overload.

SYSTEM ARCHITECTURE



DEPLOYMENT

FLINK

Configuration:

- Parallelism: 1
- Number of Task Managers: 1
- Number of Task Slots: 10

Job Manager

Task Manager

PREPROCESSING

Faust-Preprocessor

- Implemented using the **faust** library.
  - Designed for stream based applications
- Is "just" simple **Python** ☺
- Leverages the **async/await** keywords.
- Ingests preprocessed data in another Kafka Topic.

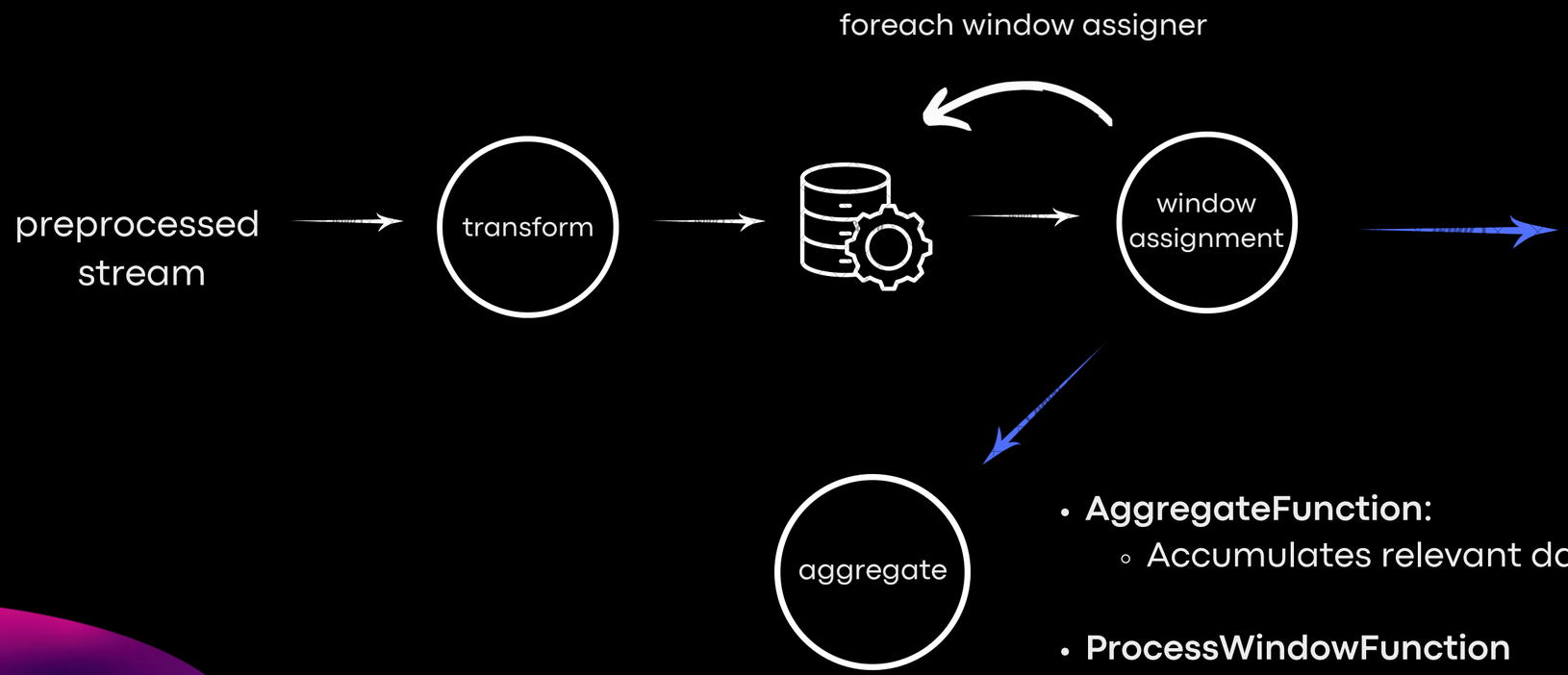
- Directly implemented in Flink using the **DataStream API** within the Python SDK.
- Prepares for processing by consuming monitoring events from Kafka.

PREPROCESSING OPERATIONS

- Select only the fileds of interest for the first **2 queries**:
  - date
  - serial_number
  - model
  - failure
  - vault_id
  - s194_temperature_celsius
- Filter out tuples where the failure, vault_id, or s194_temperature_celsius field is set to NULL
- Regex for serial numbers and models:
  - Serial numbers:
    - ^[A-Z0-9_-]+$
  - Models:
    - ^[A-Za-z0-9 _.-]+$
- Type casting
  - To ensure type safety

PROCESSING

A high level overview of the queries execution

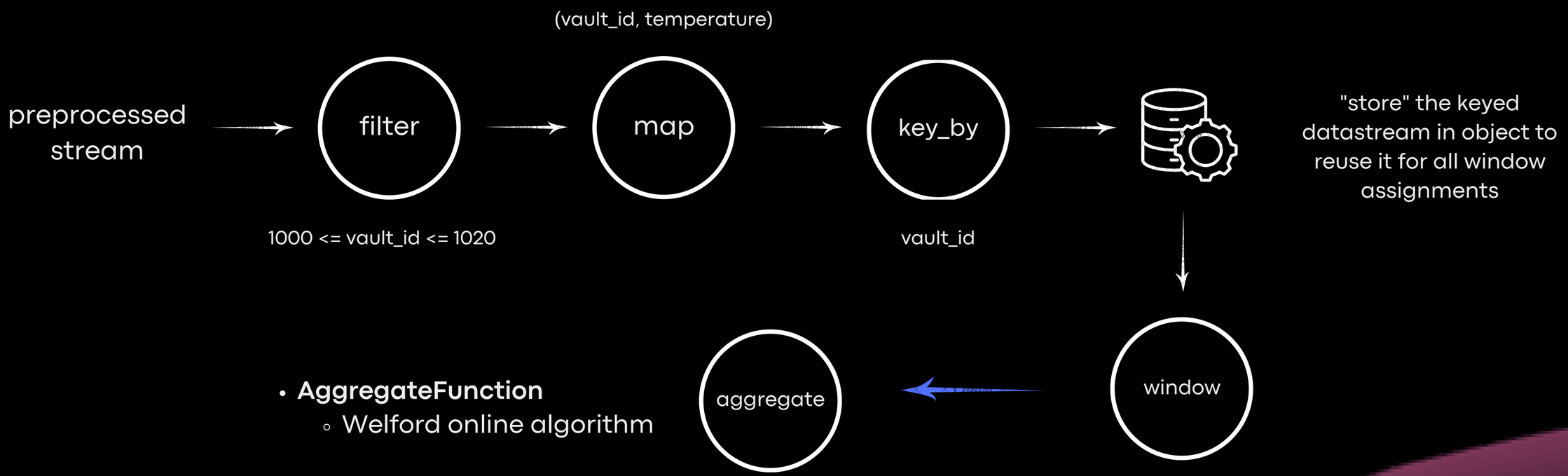


Tumbling Window assigners

- 1 Day
- 3 Days
- All days

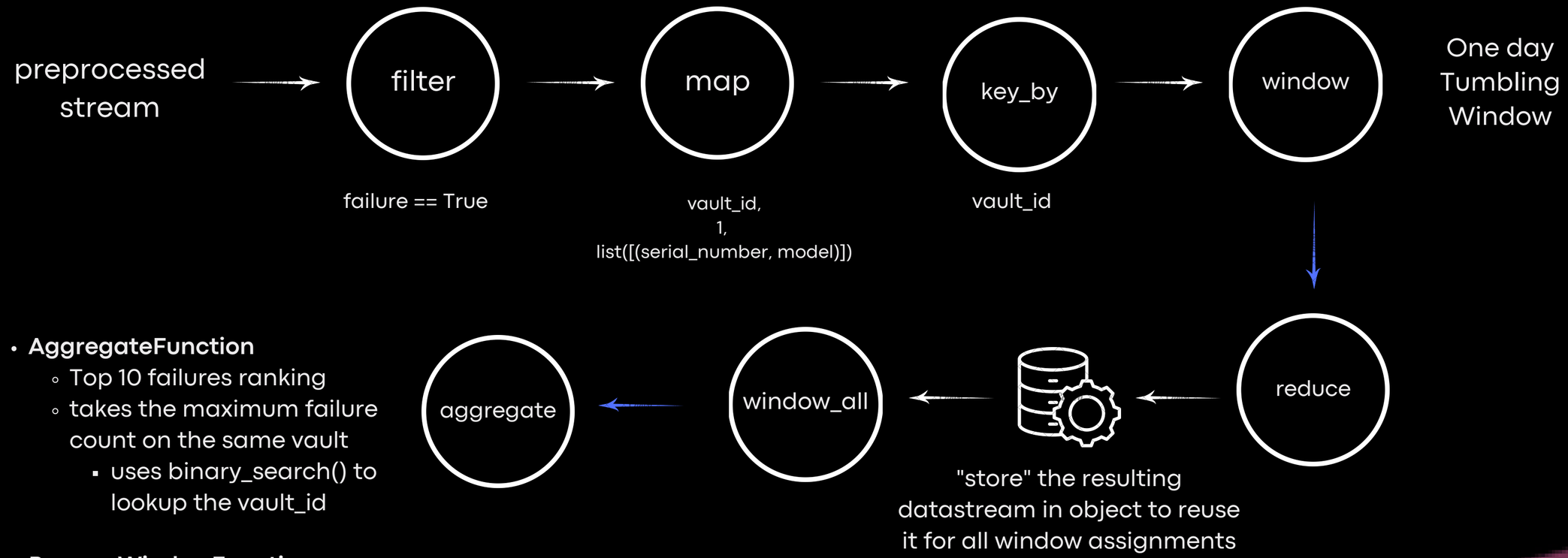
- **AggregateFunction:**
  - Accumulates relevant data
- **ProcessWindowFunction**
  - Terminates the processing over the accumulated values

QUERY 1



- **AggregateFunction**
  - Welford online algorithm
- **ProcessWindowFunction**
  - compute stddev
  - start timestamp of window

QUERY 2



- **AggregateFunction**
  - Top 10 failures ranking
  - takes the maximum failure count on the same vault
    - uses binary_search() to lookup the vault_id
- **ProcessWindowFunction**
  - **start timestamp of window**

Query limitation:
Two chained windows

METRICS EVALUATION

Throughput

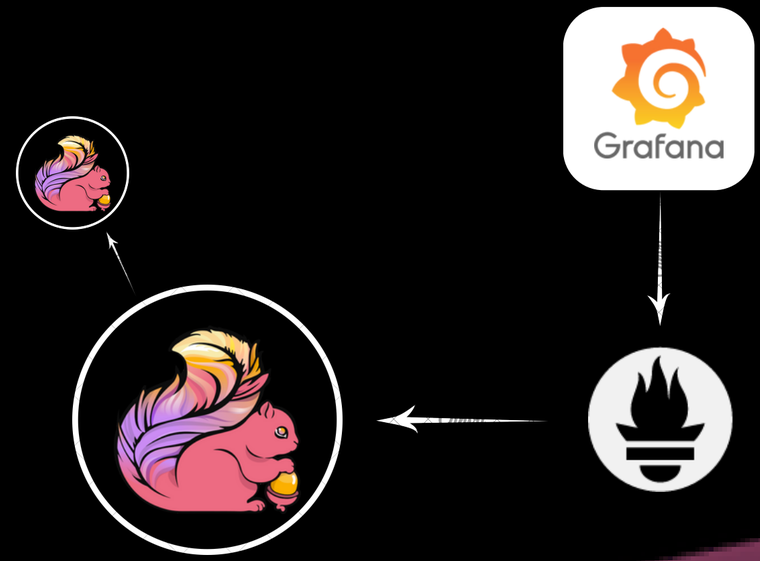
Flink's metric

numRecordsOutPerSecond

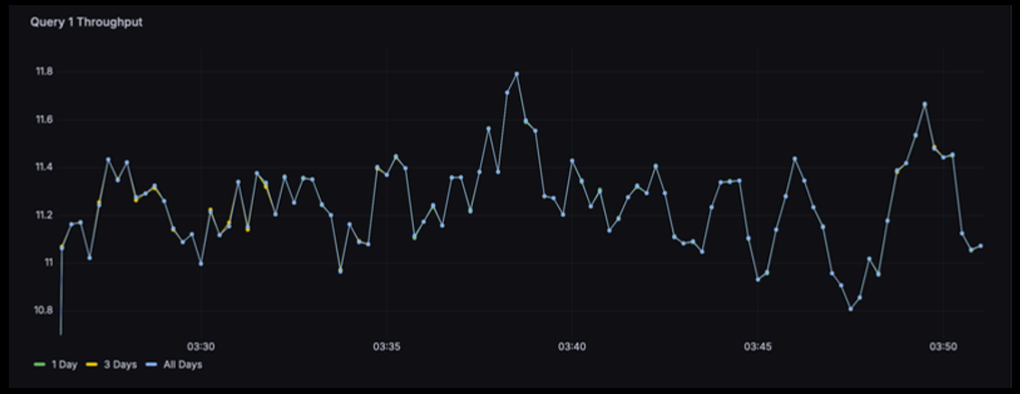
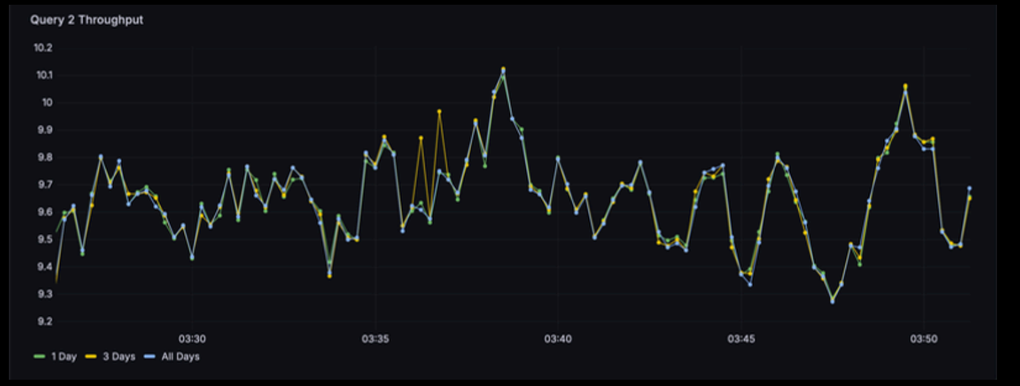
Latency

Flink's Latency Tracker

latency markers

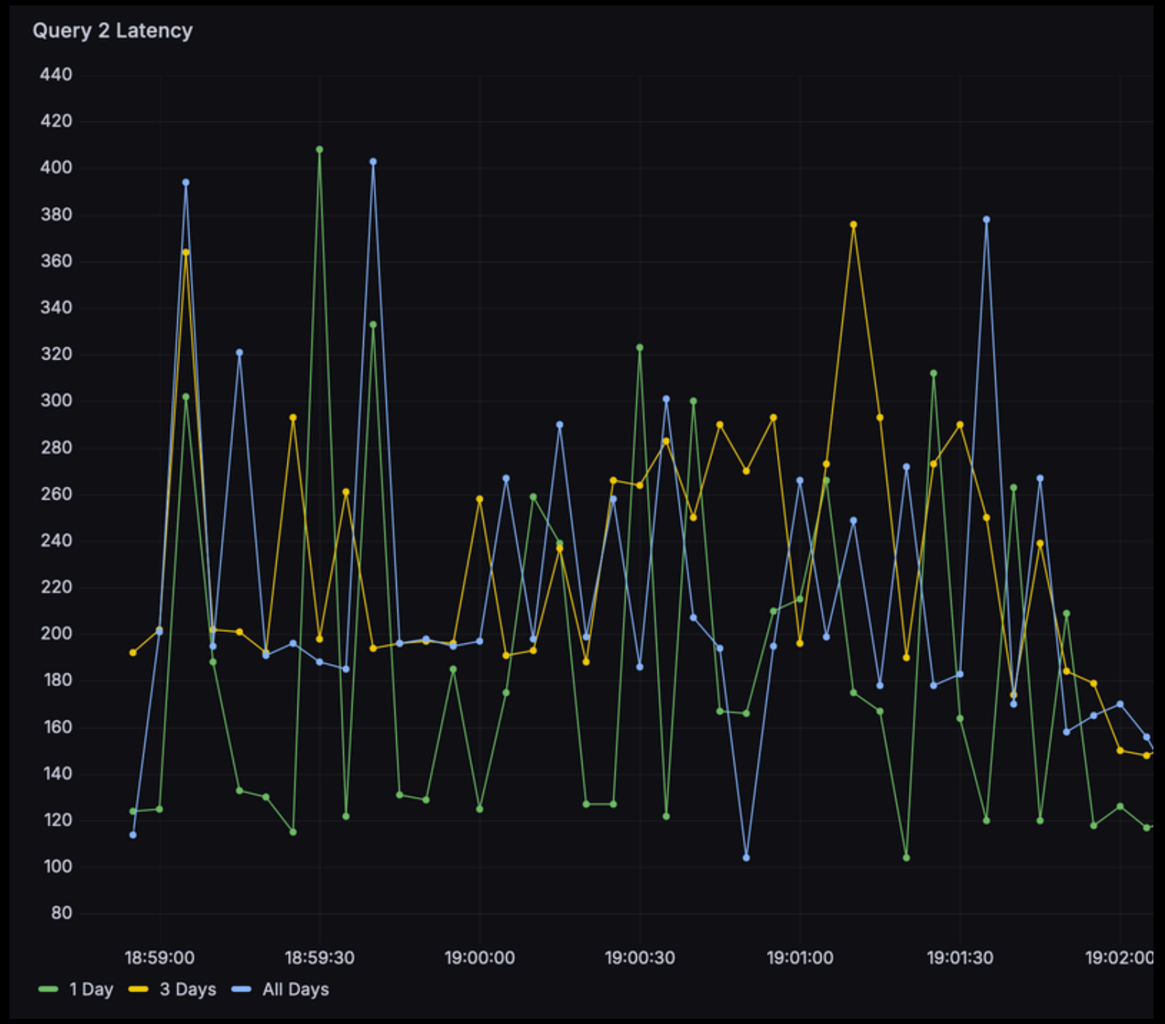


THROUGHPUT

	High arrival rate	Uniformly distributed arrivals
Query 1		
Query 2		

LATENCY

Query 1



Query 2

THANK YOU!