Time-Series Database Systems

Logan Drawdy, Mason Hudson, Amanda Lee, Zhong Zhang

Why Time-Series Databases?

- Purpose-built for handling time-series data
- Time is the key index
- Capable of handling extremely large amounts of data continuously streaming in

Scale of Time-Series Data

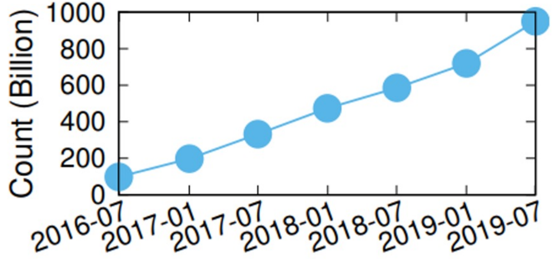


Figure 8: Time series count.

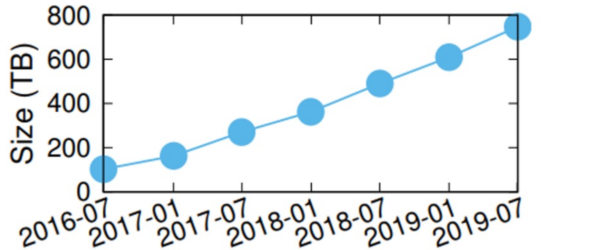


Figure 9: Time series memory size.

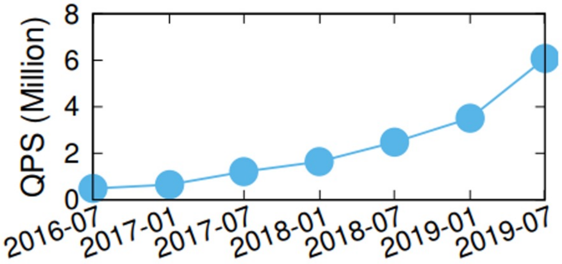


Figure 10: Queries per second.

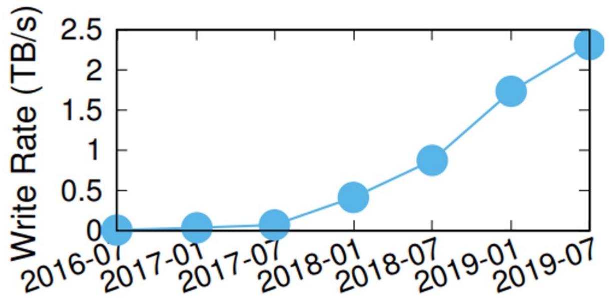
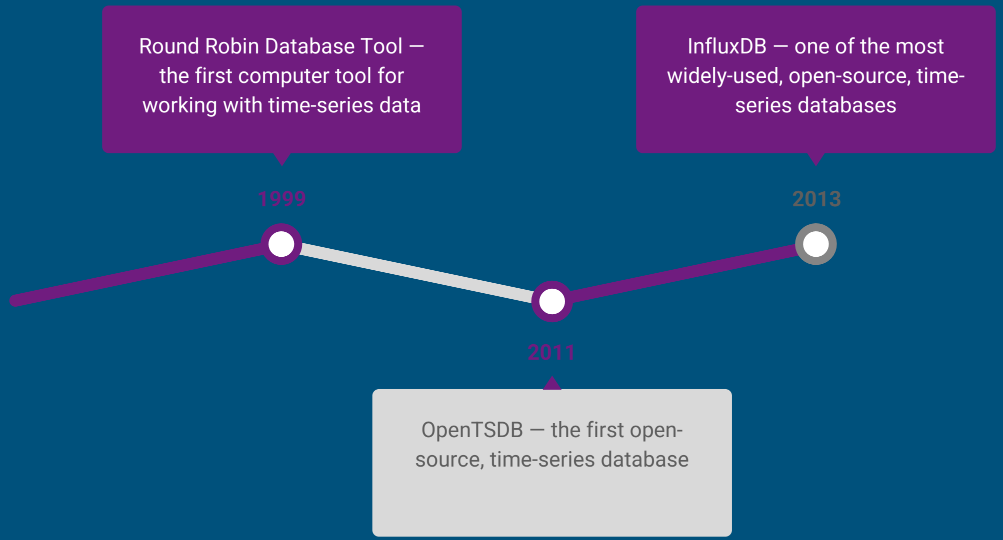


Figure 11: Time series data written per second.

https://www.vldb.org/pvldb/vol13/p3181-adams.pdf

Historical Timeline



What is Time-Series Data?

- Data that has a value and a time stamp

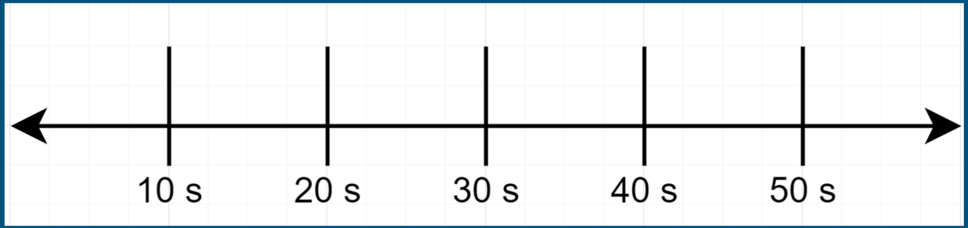
Value

The quantitative or qualitative measurement being tracked

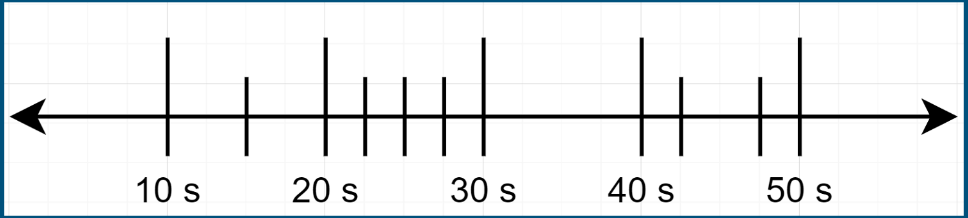
Time stamp

A precise reference for each data point in the context of time—recorded in a uniform and consistent format

What can Time-Series Data Look Like?



Values collected at equal intervals



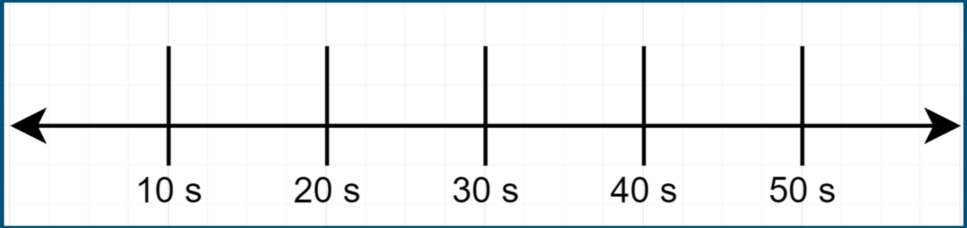
Number of events within intervals collected as values

Where Does Time-Series Data Come From?

- Typically from industrial IoT applications
- Primarily sensor data
- Think temperature, humidity, power meter data
- Continuous, time-stamped data

https://unsplash.com/photos/white-windmill-on-yellow-petaled-flower-field-during-daytime-E56cTF65xFw

Windmill Time-Series Data Example



Windmill gearbox oil temperature every 10 seconds

Timestamp	Gearbox oil temperature
2024-03-09T12:00:00Z	75°C
2024-03-09T12:00:10Z	76°C
2024-03-09T12:00:20Z	77°C
2024-03-09T12:00:30Z	78°C
2024-03-09T12:00:40Z	79°C

Windmill Time-Series Data Example

Number of hydraulic pressure sensor faults per twelve hours

Timestamp	Hydraulic pressure sensor faults
2024-03-09T12:00:00Z	0
2024-03-10T00:00:00Z	1
2024-03-10T12:00:00Z	3
2024-03-11T00:00:00Z	0
2024-03-11T12:00:00Z	2

How can all this time-series data be used?

- Trend Analysis
  - Identify long-term increases or decreases in data over time
- Pattern Recognition
  - Detect and analyze seasonal patterns, cycles, or recurring trends within the data
- Predictive Analysis
  - Predict future values or trends using historical time series data

Features and Functions

- **High Write Throughput**
  - Handle massive write loads without sacrificing read performance
- **Time-stamped Data Compression**
  - Compresses time-stamped data to reduce storage costs while maintaining query speed
- **Data Summarization**
  - Quick generation of aggregated views of data
- **Data Lifecycle Management / Retention Policies**
  - Automatically manages data retention to optimize storage and performance

Features and Functions

- **Real-time Analysis**
  - Provides the capability to perform real-time analytics and instant querying on streaming data
- **Scalability**
  - Easily scales to accommodate increasing data volumes and more complex querying demands
- **Built-in Time Series Functions**
  - Offers a suite of specialized functions for efficient time series data manipulation

Leading Products Comparison

	InfluxDB	TimescaleDB	Prometheus	Graphite
Scalability	🟢 Supports clustering	🟢 Support Clustering	🟢 Support Federation	🟠 Basic Horizontally scalable
Performance	Designed for high write & query throughput, low latency	Designed for high write and query throughput, low latency	Designed for real-time monitoring and alerting	Designed for high performance and low overhead
Data Consistency	🟠 Eventual Consistency & Support customization	🟢 Strong Consistency with ACID Compliant	🟠 Eventual Consistency	🟠 Eventual Consistency
Security	🟢 Robust	🟢 Robust	🟠 Limited (Often used in conjunction with other tools)	🟠 Limited
Community Support	🟢 Strong	🟠 Growing	🟢 Strong	🟢 Strong
Ecosystem	Wide range of plugins, Integrations & support various programming languages	Wide range of plugins, Integrations & support various programming languages	Limited to third-party tools & Primarily focused on monitoring & alerting	Wide range of plugins, Integrations & support various programming languages
Query Language	InfluxQL & Flux	SQL with extensions for time-series data	PromQL	Graphite-QL

https://blog.bytebytego.com/p/understanding-database-types

Products Technical Details

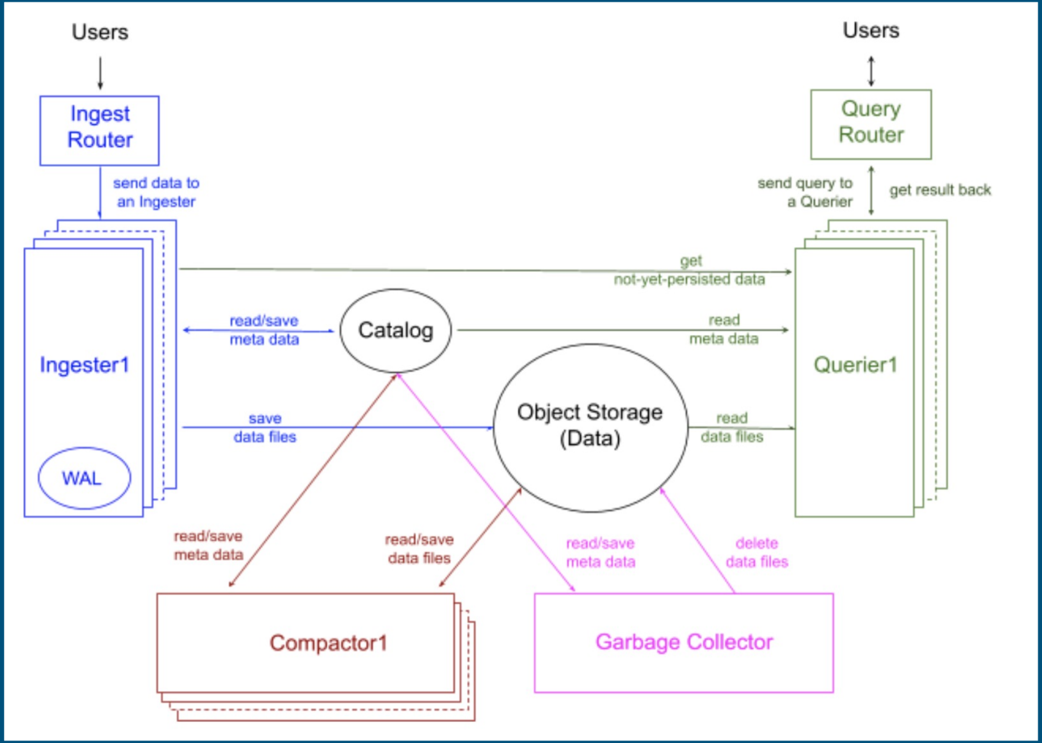
InfluxDB (NoSQL)

TimescaleDB (SQL)

https://www.niagaramarketplace.com/influxdb.html
https://iconscout.com/free-icon/timescaledb-1958407

InfluxDB - Architecture

- **Ingestion**
- **Storage**
- **Compactor**
- **Query Processing**



https://www.infoq.com/news/2023/08/influxdb-3-architecture/

InfluxDB - Modes of Operation

- **Single Node**
  - Best suited for small applications or environments with a low workload
- **Cluster**
  - Offers horizontal scalability to handle larger workloads and provides high availability for enterprise deployments.
- **Cloud**
  - A fully managed service providing scalability and reliability

InfluxDB - Security

- **Authentication**
  - InfluxDB enforces user credential checks and supports token-based access for secure API interactions.

- **Encryption**
  - It secures data in transit with TLS, protecting against eavesdropping and man-in-the-middle attacks.

- **Logs & Data Backup**
  - Maintains detailed transaction logs for audit trails and facilitates automated snapshots for reliable data recovery.

TimescaleDB - Architecture & Functions

- **On top of PostgreSQL**
  - Leverages PostgreSQL's reliability and rich feature set while enhancing time-series data management.
- **Hypertables & Chunks**
  - Utilizes hypertables to automatically partition time-series data into manageable chunks for improved performance.
- **Query Optimization**
  - Employs advanced optimization techniques to accelerate time-series specific queries and reduce latency.
- **Compression**
  - Implements sophisticated compression algorithms to minimize storage requirements and improve query efficiency.

TimescaleDB - Modes of Operation

- **Single Node**
  - Operates on a single server, suitable for development or lower-scale production environments.
- **Distributed Hypertables**
  - Scales horizontally across multiple servers for increased data ingestion and query capacity.
- **Cloud**
  - A fully managed, hosted service providing ease of use, with no maintenance overhead.

TimescaleDB - Security

- **Authentication for PostgreSQL**
  - Inherits PostgreSQL's robust user authentication system for secure access control.
- **Automatic Data Retention**
  - Configurable data retention policies automatically purge old data to maintain storage efficiency.
- **Access Control for Hypertables**
  - Access control mechanisms allow permissions to be set at the hypertable level for enhanced security.

InfluxDB Case Study 1: Capital One

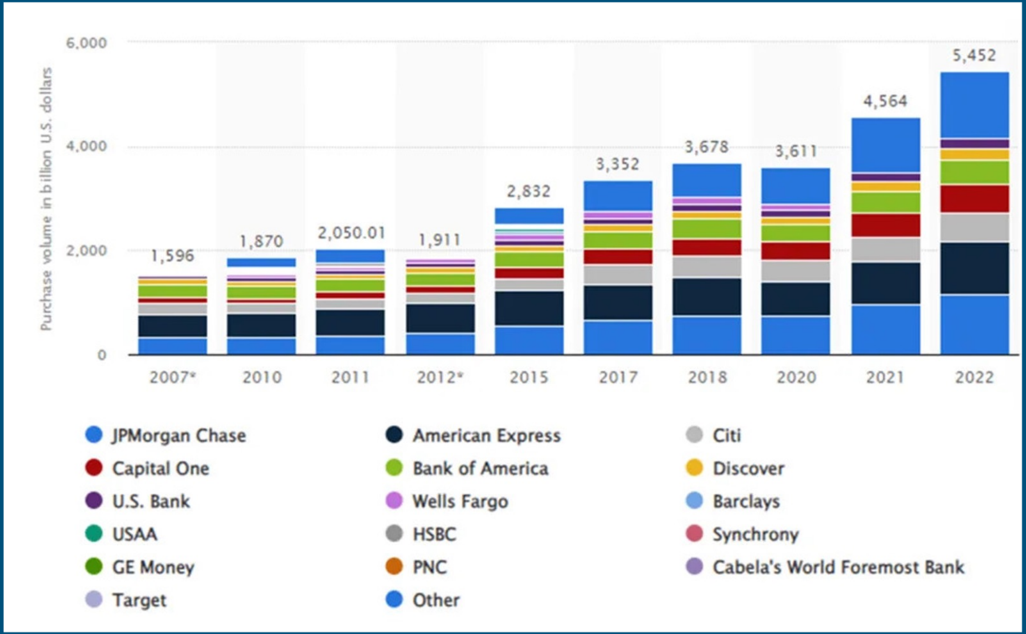
- Banking company specializing in credit, banking, and saving products

- Have many metrics on infrastructure and business applications that all change over time

https://en.m.wikipedia.org/wiki/File:Capital_One_logo.svg

Metrics Requiring Measurement

- User volume changes and transaction tracking
- CPU and memory usage in internal servers
- Application and Other Database metrics



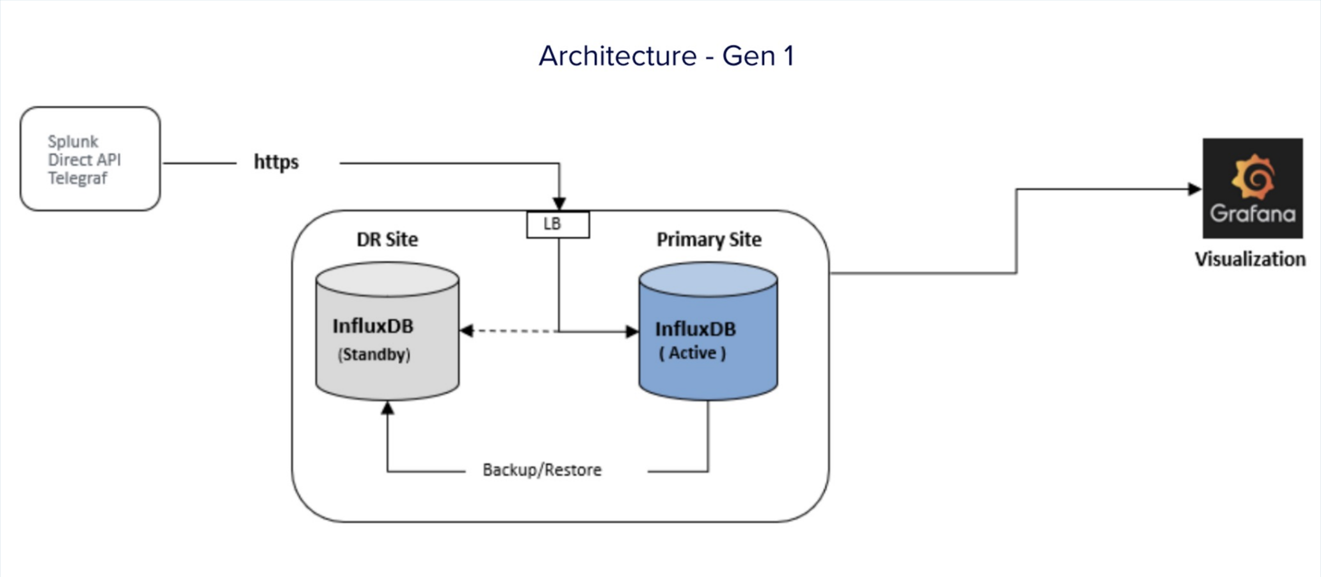
https://finance.yahoo.com/news/capital-one-trying-become-next-134600233.html

System Requirements:

- Achieve resilience and protect time-series data
- Create high data availability for customers to access their metrics
- Use data in ml models to forecast potential vulnerabilities and respond accordingly

- The solution was to use influxDB and exploit its high speed read and write to almost instantaneously view and model data

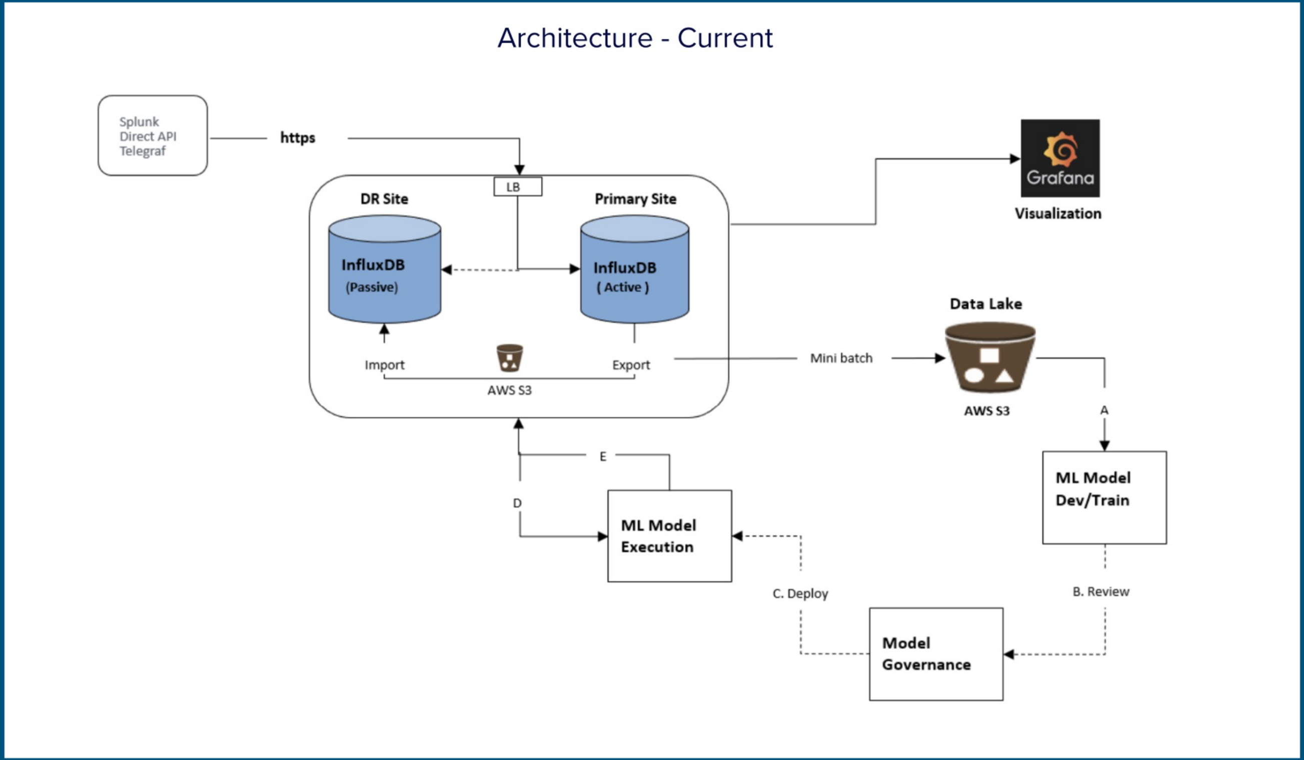
System One:



https://get.influxdata.com/rs/972-GDU-533/images/Customer_Case_Study_Capital_One.pdf

- 80/20 Rule
- Backup issues

Current System:



https://get.influxdata.com/rs/972-GDU-533/images/Customer_Case_Study_Capital_One.pdf

InfluxDB Case Study 2: Robinhood

- Robinhood is a pioneer of commission free investing, intending to make investing accessible to more people.

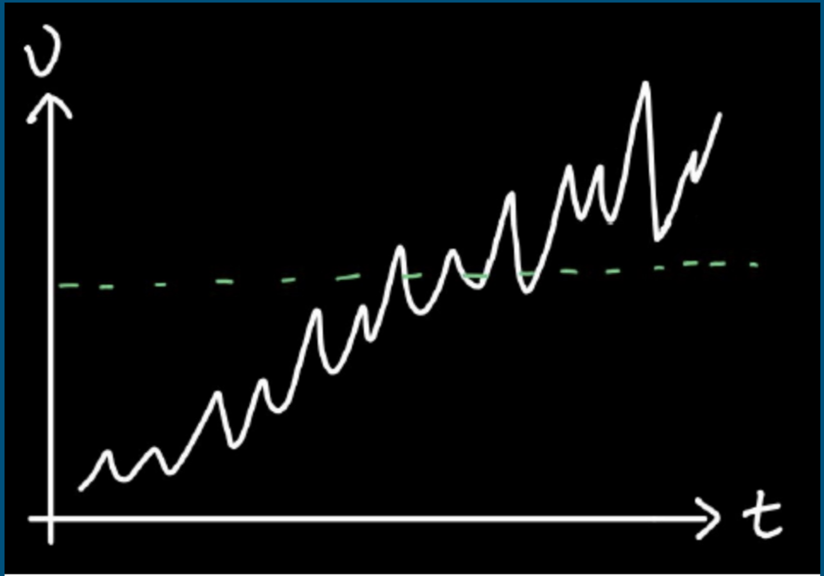
- Internally to do this Robinhood needed to understand their risk vectors and mitigate them

- As the number of time series grow, the amount of effort required to detect and understand anomalies becomes increasingly costly

https://commons.wikimedia.org/wiki/File:Robinhood_Logo.png

System Requirements

- The solution was to build and automated anomaly detection system
- To do this they developed a ml model to determine adequate thresholds for price anomalies using historical price data of stocks.
- In order to do this they needed a real time querying system with fast ingestion and aggregation, as well as alerting capabilities.



Threshold-based alerting

https://get.influxdata.com/rs/972-GDU-533/images/Customer_Case_Study_Robinhood.pdf

Why InfluxDB

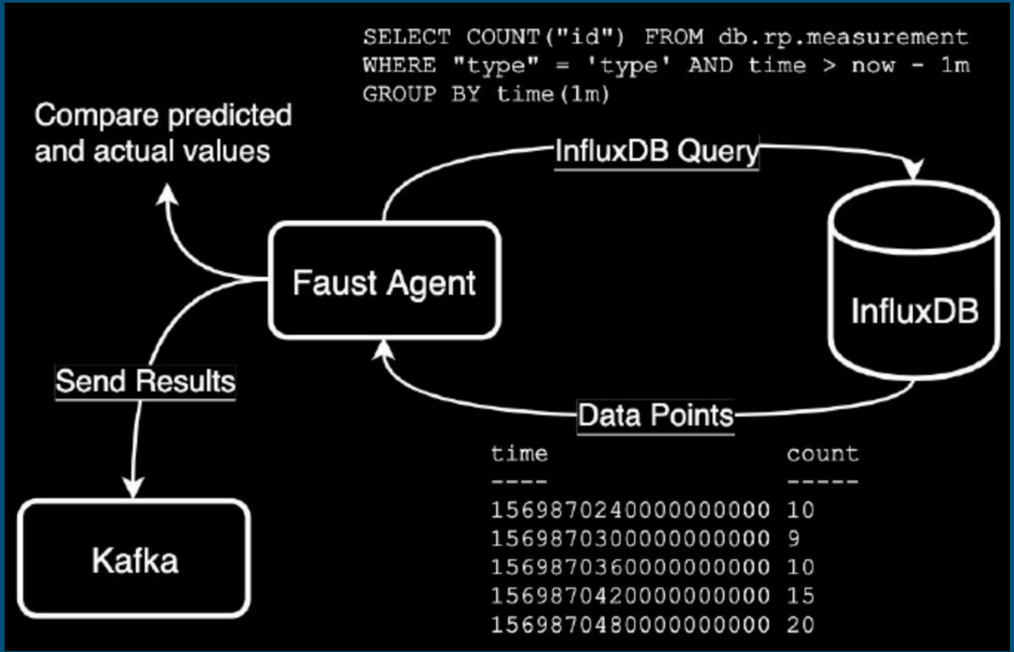
- lightweightness (doesn’t require third parties to run)
- the fact that it is schemaless which reduces overhead
- Indexing via specific fields in data
- And fast read and write capabilities

The System

- They then needed to create a real time stream processing system, which they used pythons version of kafka streams for

- This allowed them to retrieve data and run their algorithm on it continuously in real time.

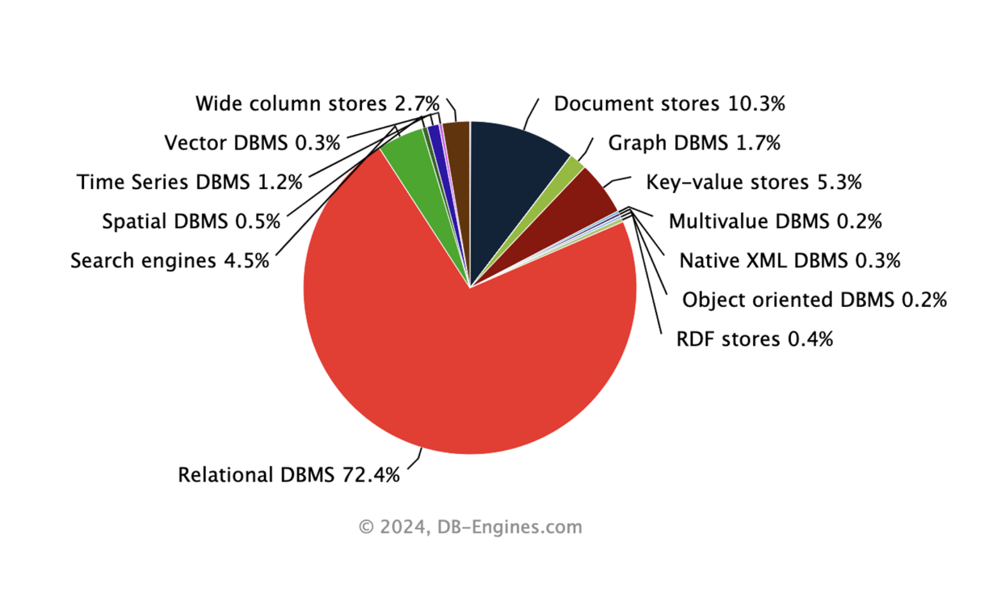
- They then compare their predicted results and actual values and then alert for anomalies



https://get.influxdata.com/rs/972-GDU-533/images/Customer_Case_Study_Robinhood.pdf

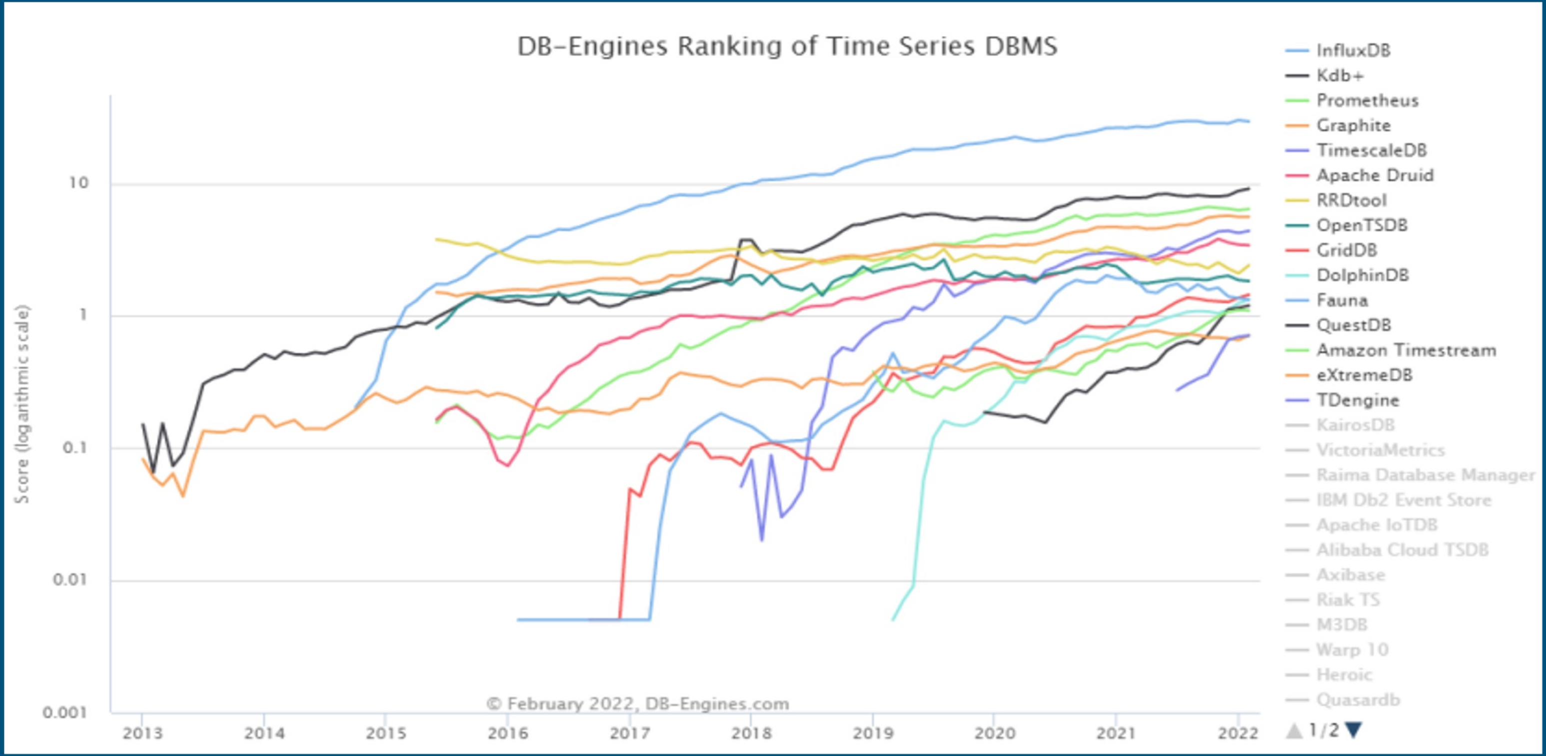
The Market

- According to Verified Market Research, the Global Time Series Databases Software Market was valued at USD 273.56 Million in 2020 and is projected to reach USD 575.03 Million by 2028, growing at a CAGR of 10.06% from 2021 to 2028, particularly attributed to growth in the IoT market.



https://db-engines.com/en/ranking_categories

The Market



https://db-engines.com/en/ranking/time+series+dbms

Prognosis

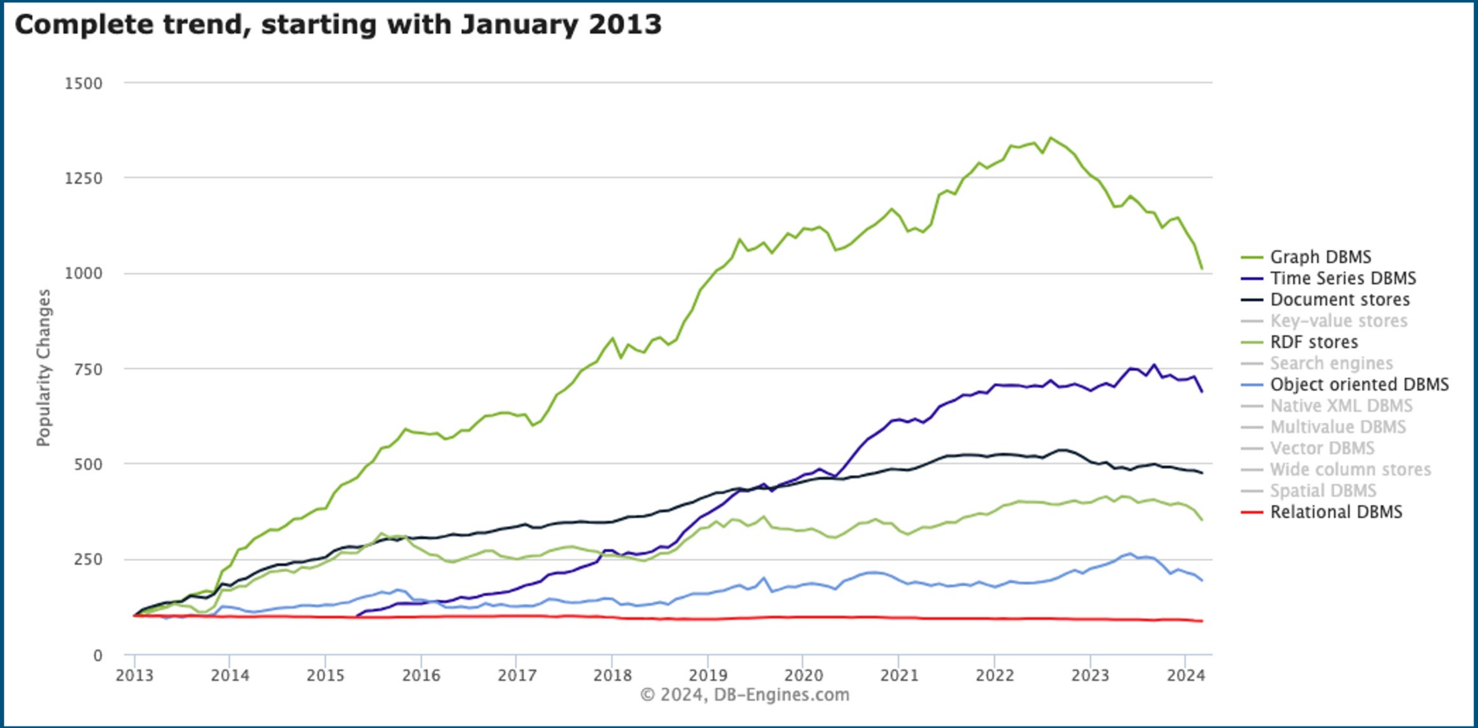
Current Goal

Find ways to handle bigger data streams with more complicated analytics without sacrificing running efficiency.

Long-term Concerns

Turning high-resolution data into lower-resolution, historical summary to save space, while still providing useful summaries.

Prognosis



https://db-engines.com/en/ranking_categories

Prognosis

Increased supply of time series data → Constant innovation for optimizing big data intake

Tight relationship with analytics → Efficient integration of TSDB and other technologies

Expansion and specialization of TSDB

Current Research

Challenges:

- Structural and functional aspects
  - Fitting into snapshots
  - Defining temporal relations
  - Organizing time series into arbitrary groups

Potential areas of improvement:

- Handling continuous stream of data
  - Batch learning of models not scalable -> sequential learning algorithms
  - Design of efficient models to handle updates

Current Research

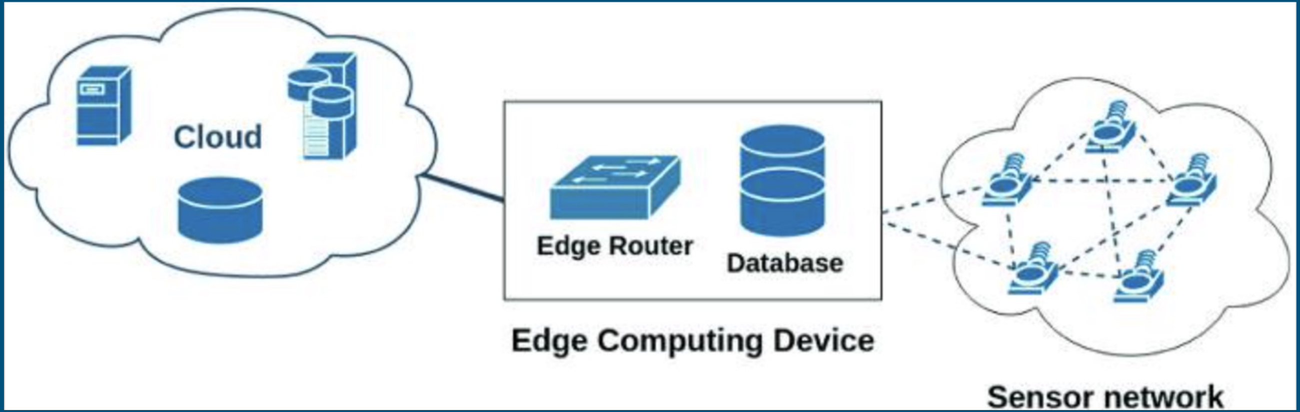
- Time-series in ML
  - Cold start problem
  - Synthetic data generators
  - Transfer learning
  - Non-homogeneity even within same domain

- Multivariate modeling
  - Variability in the time scales
  - Dynamic time warping comes with computational cost and loss of accuracy

Recent Research Papers

1. Comparative analysis of time series databases in the context of **edge computing for IoT**

- Performance measured by execution time
- Testing on insertion and querying operations
- PostgreSQL and InfluxDB best performed for reading data
- PostgreSQL best performed for insertion

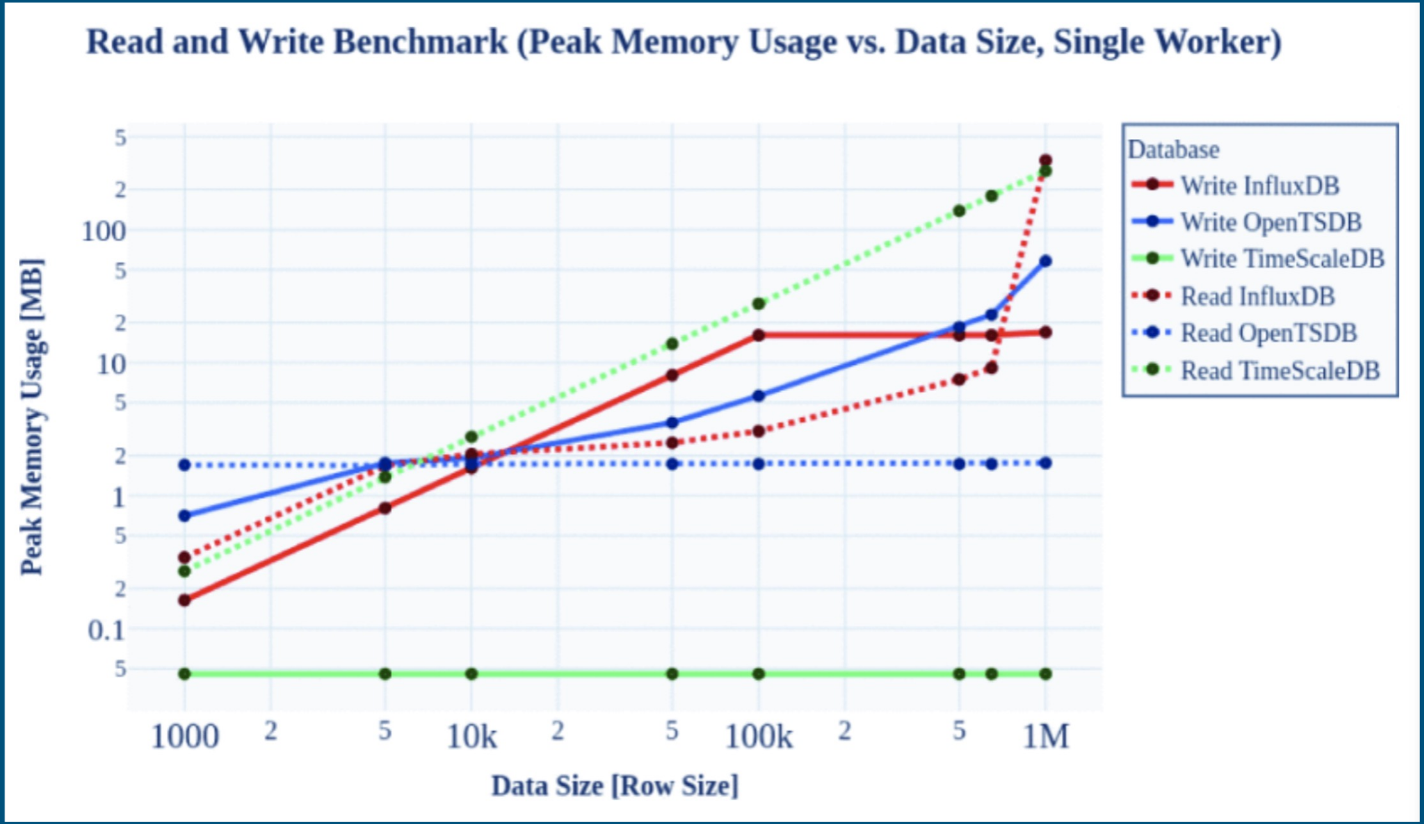


https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7302557/

Recent Research Papers

2. TSDB benchmark framework to compare 3 TSDB systems for **power measurement data**

- Different database options may be more suitable for certain data domains
- Performance based on execution time, memory consumption, and throughput
- Performance benchmarks for writing, concurrency, and query executions



https://ieeexplore.ieee.org/document/9559822

Q&A