



# Time series databases: Study & Benchmarking

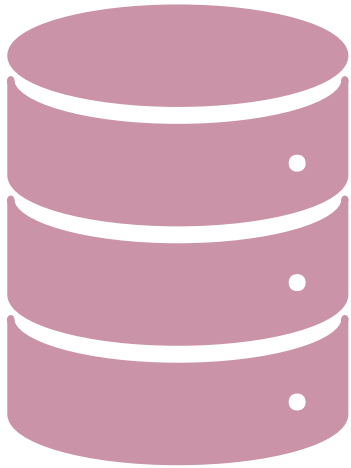
VASILEIOS ILIAS DROUZAS

SUPERVISOR: PROF. YANNIS KOTIDIS

This presentation is part of my BSc Thesis in Time Series Databases  
Important part of the research was held in [ISDB](#) lab.

# Contents

- What is time series data
- Studying time series databases (TSDBs)
  - ✓ InfluxDB
  - ✓ TimescaleDB
  - ✓ QuestDB
  - ✓ Prometheus
- Benchmarking TSDBs
  - ✓ Head to head performance
  - ✓ Cardinality impact
  - ✓ Multi-threading impact
- Materialized views



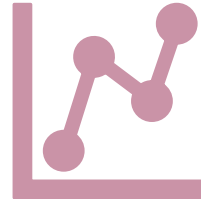
# What is time series data?

---

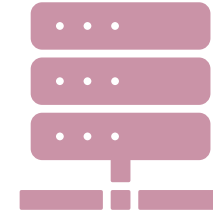
# Time series data



Observations obtained through time (e.g. health metrics)



Can be distinguished in univariate (single observations over time) and multivariate (2+ variables)

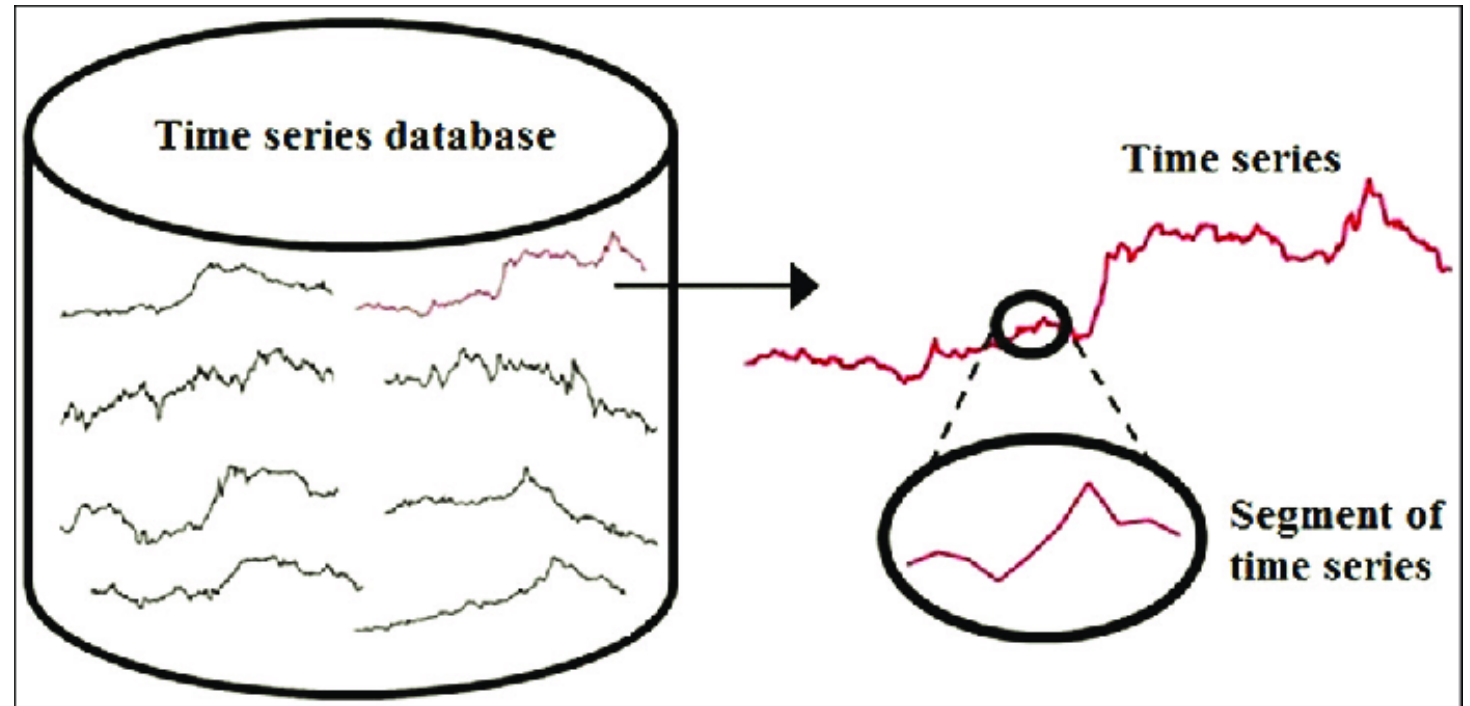


Used in forecasting, clustering/classification, signal detection/estimation

# TSDBs

---

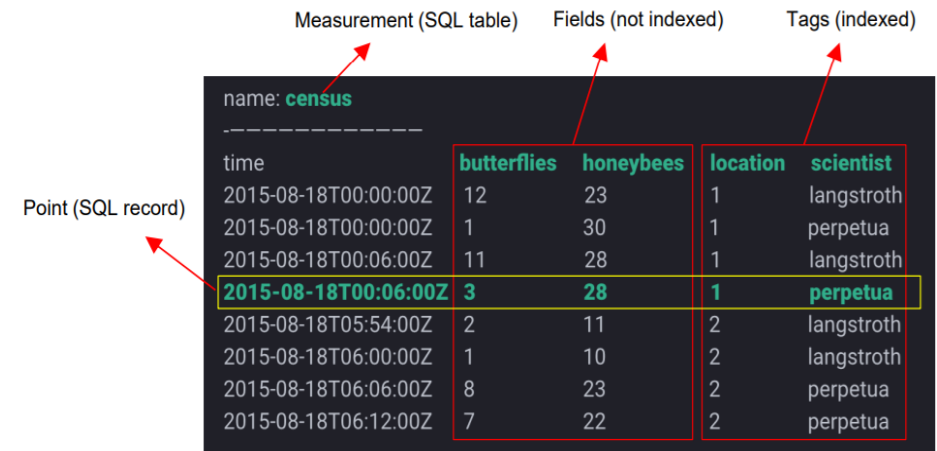
- Time series databases
- Time is treated as a first-class citizen (and not an extra field)
- Use of continuous queries, retention policies etc.
- These databases aim to maximize performance and query capabilities for time series data



# Studying TSDBs

# InfluxDB (1/3)

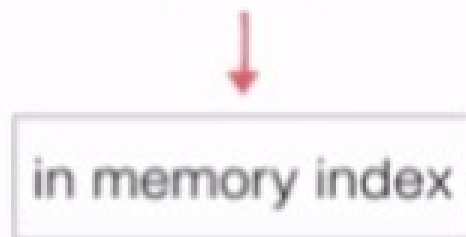
- Open - source time series database
- Written in Go
- Supports the InfluxLineProtocol
- Two query languages: InfluxQL (SQL-like), Flux
- Data elements: Timestamp, Measurement, Fields, Tags, Point
- Downsampling: Reduce disk usage, improve query performance. Achieved by continuous queries



The diagram illustrates the InfluxDB data structure with annotations. A red arrow points from 'Measurement (SQL table)' to the 'census' measurement name. Another red arrow points from 'Fields (not indexed)' to the 'butterflies' and 'honeybees' columns. A third red arrow points from 'Tags (indexed)' to the 'location' and 'scientist' columns. A fourth red arrow points from 'Point (SQL record)' to a specific row in the table.

time	butterflies	honeybees	location	scientist
2015-08-18T00:00:00Z	12	23	1	langstroth
2015-08-18T00:00:00Z	1	30	1	perpetua
2015-08-18T00:06:00Z	11	28	1	langstroth
2015-08-18T00:06:00Z	3	28	1	perpetua
2015-08-18T05:54:00Z	2	11	2	langstroth
2015-08-18T06:00:00Z	1	10	2	langstroth
2015-08-18T06:06:00Z	8	23	2	perpetua
2015-08-18T06:12:00Z	7	22	2	perpetua

awesome time series data



# InfluxDB storage system

---



# InfluxDB (3/3)

---

- Strong read/write throughputs (+)
- SQL-like query language (+)
- Connection with Grafana (+)
- Support of multiple programming languages (Java, R, Python, Ruby, Scala etc.) (+)
- Scalability offered as a close-source feature (-)
- Not a CRUD database, more like CR-ud (-)



# TimescaleDB (1/2)

---

- Open – source time series database
- Constructed on PostgreSQL, supports SQL
- Wide-column based
- A group of chunks makes a hypertable
- When partitioning, rows with the same hour timestamp are placed on the same chunk
- Chunks created automatically when adding new rows



# TimescaleDB (2/2)

---

- Continuous aggregates help boost performance, similarly to InfluxQL's continuous queries
- Retention policies
- Data model: wide-table (used also in relational databases), narrow-table (each metric combination used as an individual time series)



# QuestDB (1/3)

---

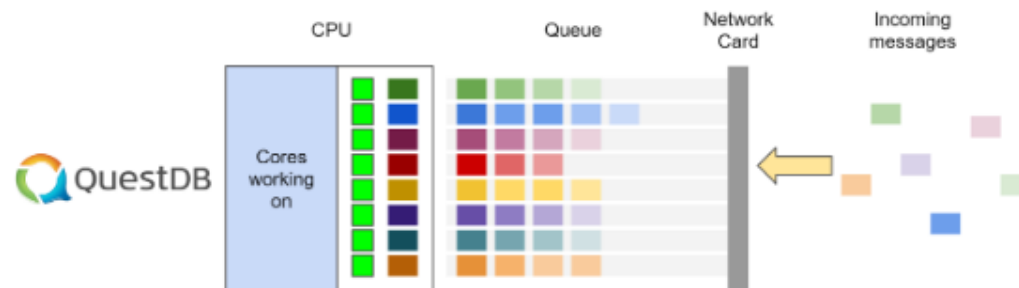
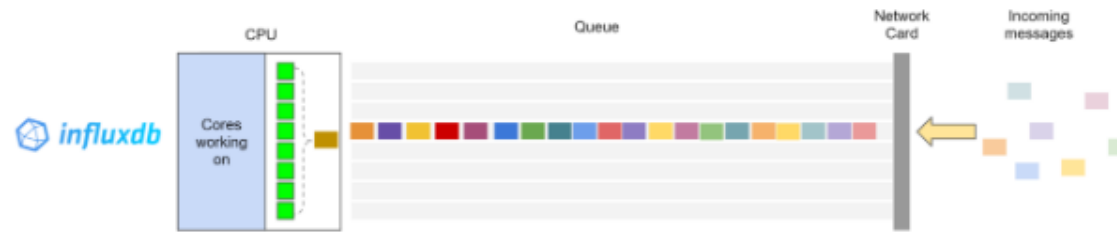
- Open – source time series database
- Includes the InfluxDB line protocol
- Implemented with SQL
- Supports partitioning, indexing
- Uses SIMD operations in query execution
- Reported to be the fastest time series database



# QuestDB – InfluxLineProtocol (2/3)

---

- QuestDB can ingest data through the InfluxLineProtocol (ILP) to take advantage of SQL to query Influx data but keeping at the same time the flexibility of ILP.
- InfluxDB faces problems , cannibalizing the CPU when dealing with big cardinality rates.
- QuestDB maximizes the utilization of the CPU, while it does not stay idle.
- QuestDB can work in parallel, InfluxDB is limited to single receiver throughput.



# QuestDB - Schemaless ingestion (3/3)

---

- QuestDB uses the InfluxLineProtocol (ILP) to ingest data without having to worry about updating the schema when needing to insert new tags and values from the measurements.
- This means QuestDB can also be schema – free.
- InfluxDB uses the ILP in the same way to benefit from schemaless ingestion.
- However, this feature is not present in TimescaleDB and most other time series databases.

# Prometheus

---

- Open - source monitoring & alerting system
- Multi-dimensional data model
- Time series collection achieved through a pull model over HTTP
- PromQL as the query language
- Alerting services (similar to triggers in vanilla SQL but more extended)



# Benchmarking TSDBs

---

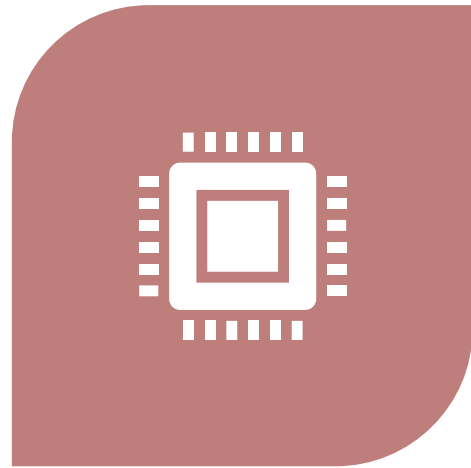


# Time Series Benchmark Suite (TSBS)

- Open-source benchmarking tool.
- Two kinds of loads: DevOps and IoT data.
- IoT data is closer to real life, containing missing & out-of-order entries.
- Data randomly generated, and then will be tested in load and query execution performance.
- *scale* flag: adjusts the number of hosts (DevOps data) or the number of trucks tracked (IoT data).

# DevOps vs IoT

---



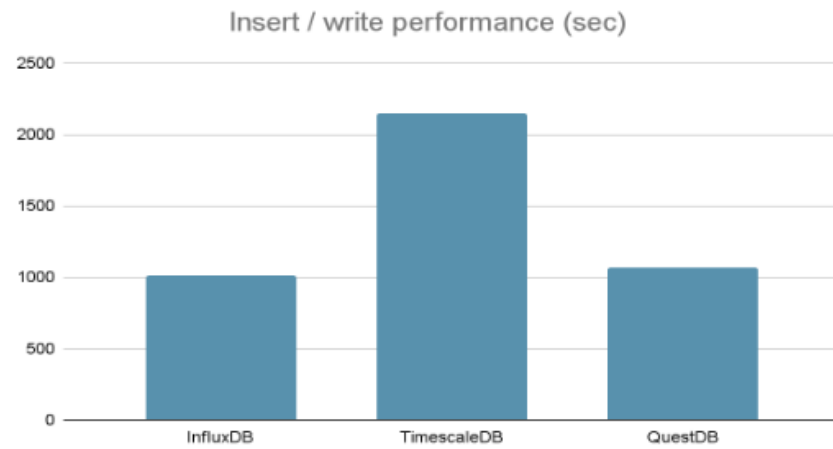
DEVOPS: GENERATE, INSERT DATA FROM 9 SYSTEMS. THE SYSTEMS GENERATE 100 METRICS PER READING INTERVAL.



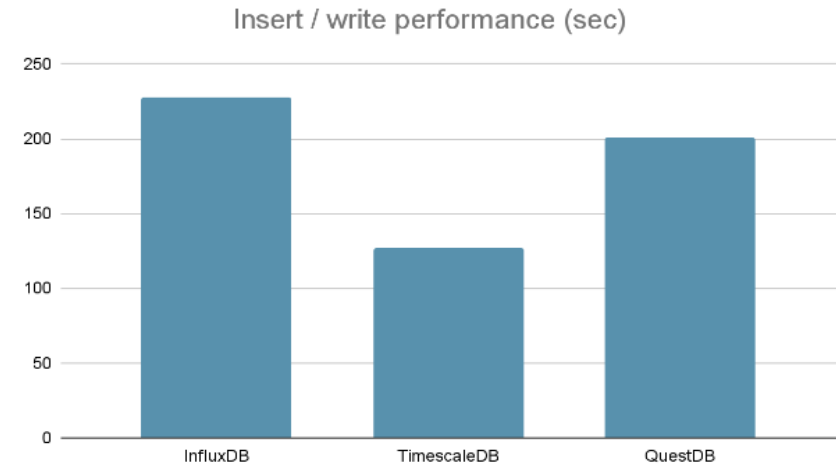
IOT: SIMULATES DATA STREAMING FROM A SET OF TRUCKS.

# Ingestion performance

---



DevOps data



IoT data

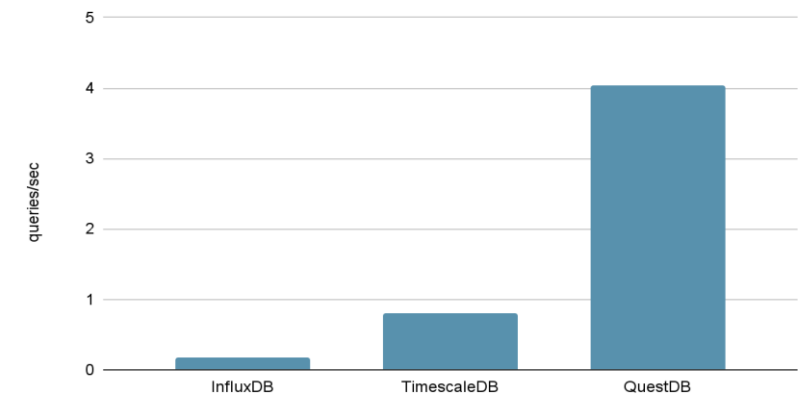
# Query execution performance (1/2)

---

Testing the execution performance of the databases based on specific queries:

1. Query type “high-cpu-all”: Returns all the readings where one metric is above a threshold across all hosts.

Query execution rates ("high-cpu-all")

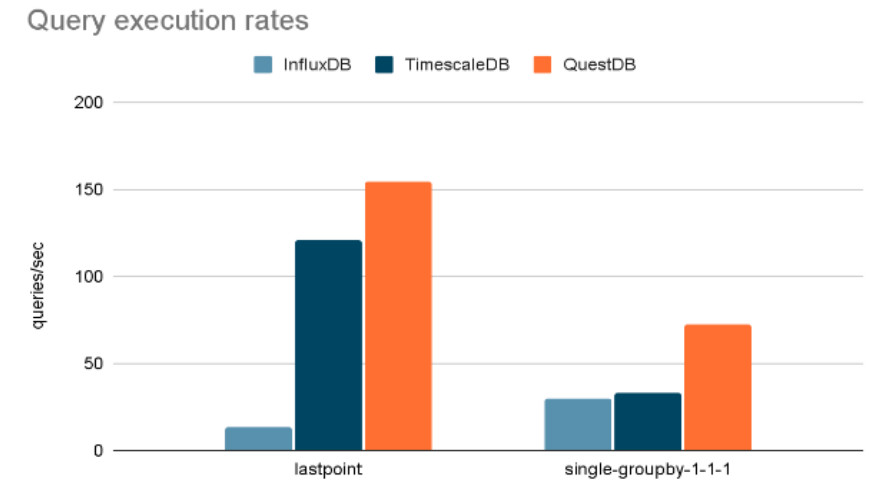


# Query execution performance (2/2)

---

Testing the execution performance of the databases based on specific queries:

2. Query type “lastpoint” : returns the last reading for each host.
3. Query type “Single-groupby-1-1-1” : a simple aggregation (MAX) on one metric for one host every five minutes for an hour.



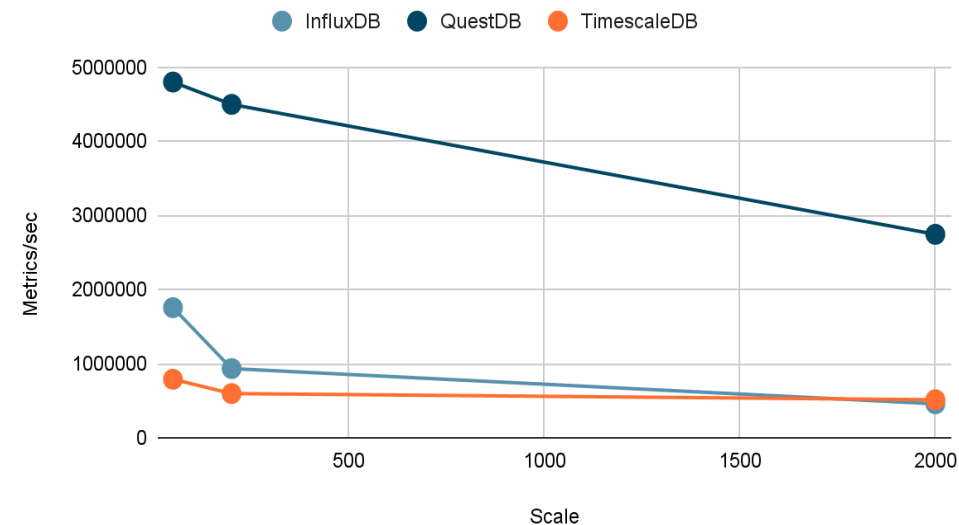
# Cardinality impact (1/2)

---

**High cardinality:** Each indexed column in a table has many unique values.

## I. Impact on ingestion rates

Insertion performance (DevOps data)



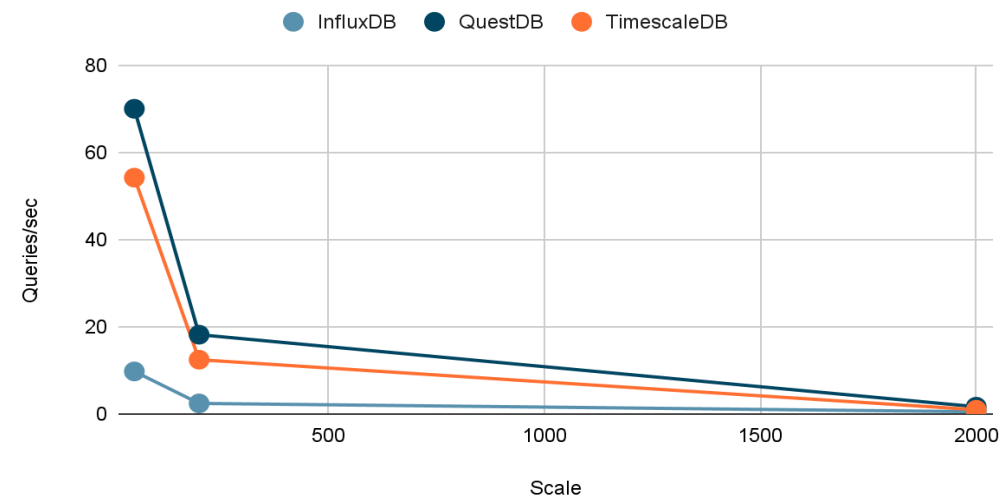
# Cardinality impact (2/2)

---

**High cardinality:** Each indexed column in a table has many unique values.

## II. Impact on query execution rates

Query execution performance ("high-cpu-all")

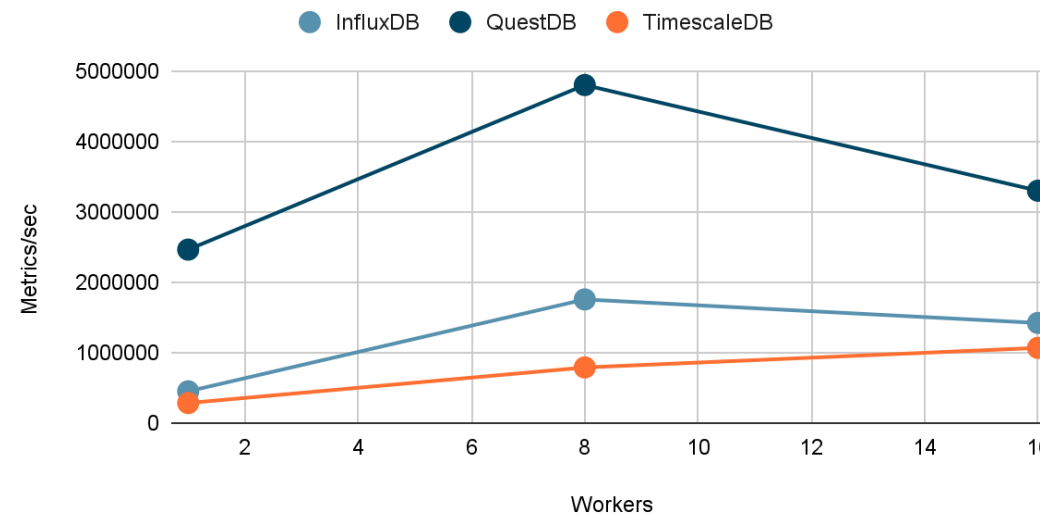


# Multi-threading impact (1/2)

---

## I. Impact on ingestion rates

How ingestion is affected due to different number of threads  
(DevOps data)

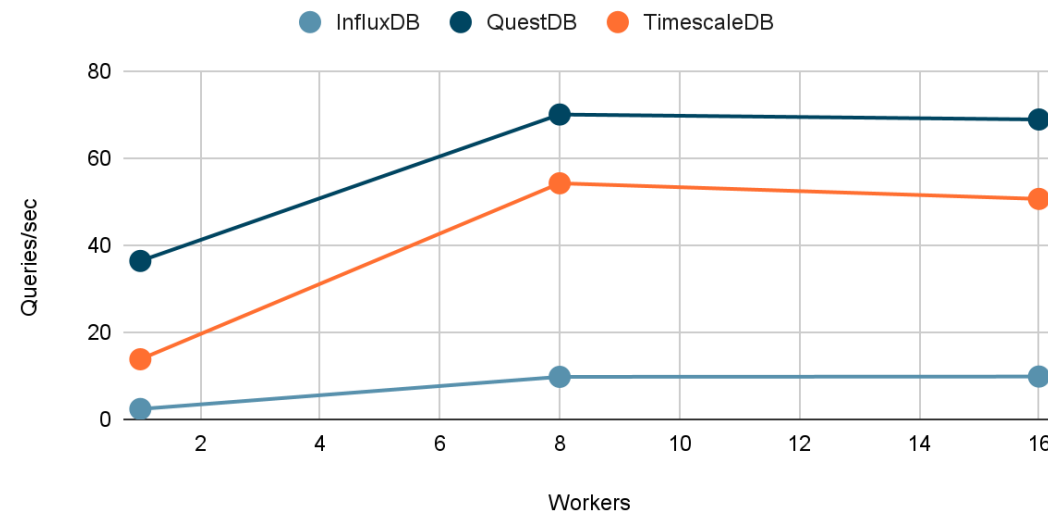




# Multi-threading impact (2/2)

## II. Impact on query execution rates

How query performance is affected due to different number of threads ("high-cpu-all" query, DevOps data)



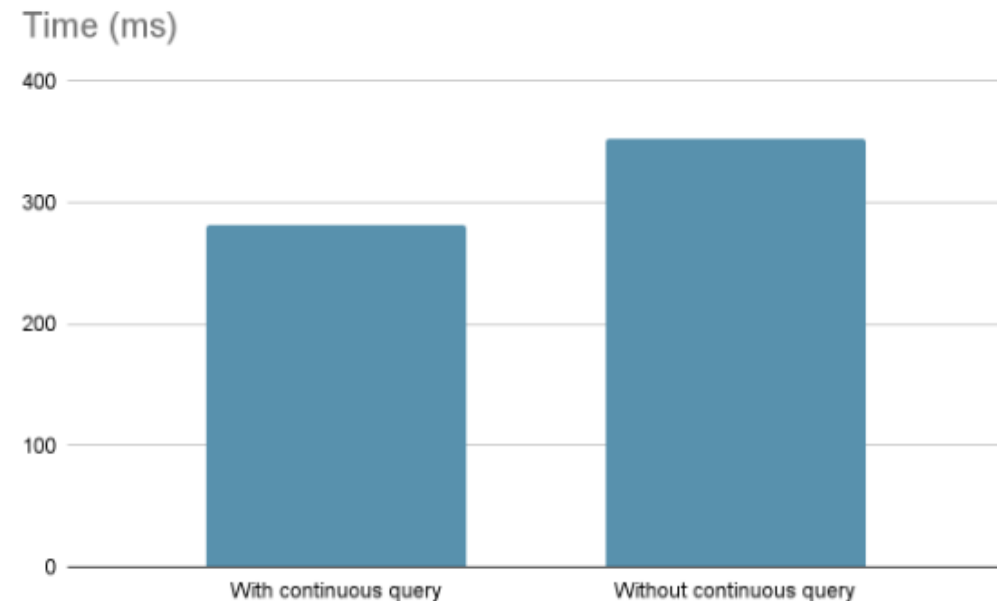
# Materialized views (1/3)

---

- In TSDBs, downsampling is supported by continuous aggregates (TimescaleDB) and continuous queries (InfluxDB).
- We will test performance on aggregate and join queries. (InfluxQL though does not support joins)

InfluxDB performance on an aggregate query is shown on the right.

InfluxDB - Aggregate query



Original size: 5 MB  
Reduced size: ~4 MB (~ -20%)

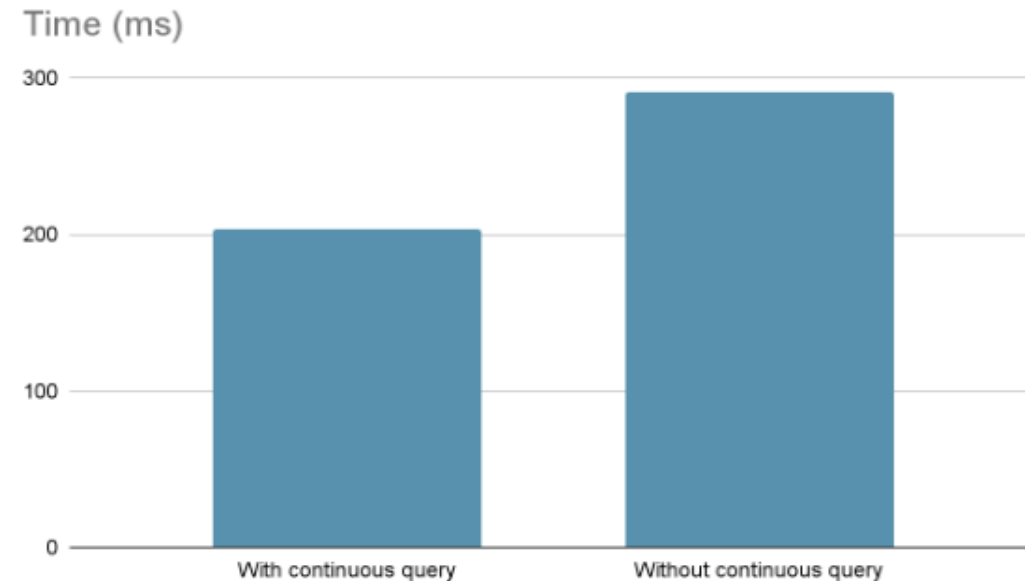
# Materialized views (2/3)

---

- In TSDBs, downsampling is supported by continuous aggregates (TimescaleDB) and continuous queries (InfluxDB).
- We will test performance on aggregate and join queries. (InfluxQL though does not support joins)

TimescaleDB performance on an aggregate query is shown on the right.

TimescaleDB - Aggregate query



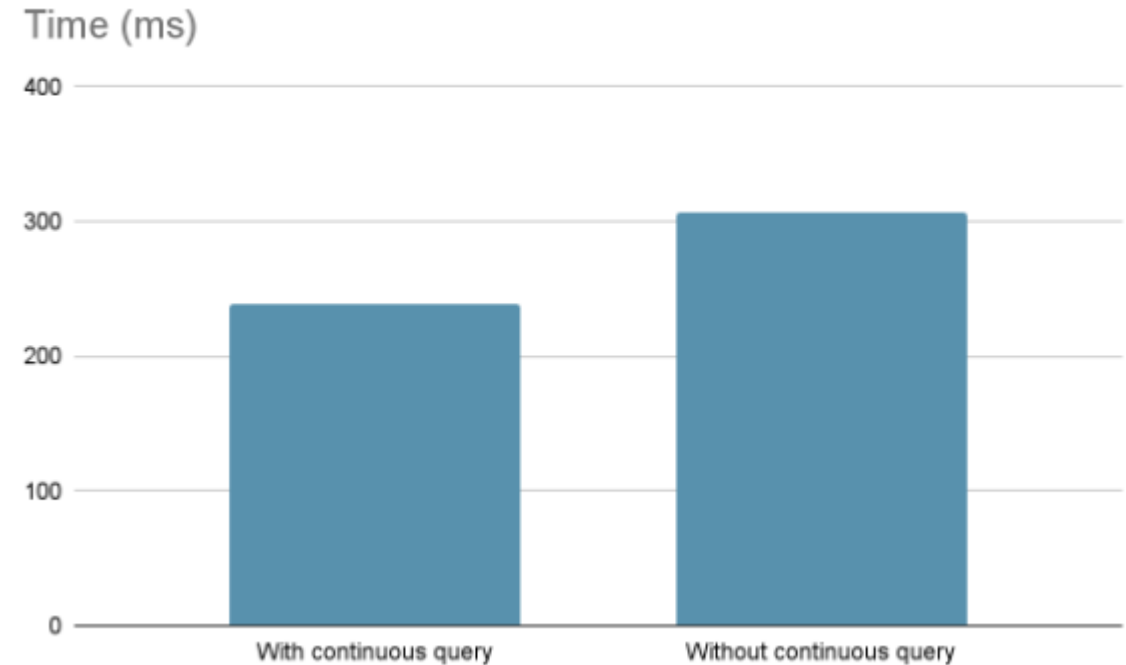
Original size: 5 MB  
Reduced size: ~3.5 MB (~ -30%)

# Materialized views (3/3)

- In TSDBs, downsampling is supported by continuous aggregates (TimescaleDB) and continuous queries (InfluxDB).
- We will test performance on aggregate and join queries. (InfluxQL though does not support joins)

TimescaleDB performance on a join query is shown on the right.

TimescaleDB - Join query



Original size: 5 MB  
Reduced size: ~3.77 MB (~ -24.5%)



Thank you!

---

# References

---

1. <https://thecustomizewindows.com/2019/10/what-is-time-series-database-tsdb/>
2. <https://github.com/questdb/questdb>