

Frank Winkler

InfluxDB vs TimeScaleDB

NHR Monitoring Workshop 2022
07/18/2022 to 07/19/2022, Erlangen

- Open-source **time series database** (TSDB) with a large community
- Implemented in Go
- Storage Engine: Time-structured merge (TSM) trees
- An entry is identified by a **measurement name**, a **timestamp**, a **field name** and **optional tags**
- Uses line protocol to write data points:

```
<measurement>,<tag set> <field set> <timestamp [s]>
```

- It provides two different languages:
 - **InfluxQL** (SQL-like query language)
 - **Flux** (functional data scripting language from version 2.0)
- Access points: HTTP API, JSON over UDP (InfluxDB-Python)

Sources: docs.influxdata.com/influxdb

InfluxDB – Write Performance



- **Stress tests with PIKA metrics** to determine the maximum number of nodes InfluxDB can handle without data loss
- Each metric source is mapped to a measurement
- Each metric value is tagged with timestamp, hostname as well as CPU/GPU
- Single database, **InfluxDB 1.7.10**, Micron M510 SSD
- Each simulated node generates about **106 values per measurement interval**
- Each experiment ran two hours

Transmission Interval	Batch Size	Simulated Nodes	Protocol
2s	106	7200	HTTP / UDP
4s	212	10,800	HTTP /UDP

- Almost no difference between HTTP and UDP in terms of write performance

- Open-source **relational database** based on PostgreSQL
- Supports standard SQL queries
- Additional **SQL functions and table structures for time series data**
- Written in C
- Storage Engine: B+ trees
- Access points: ADO.NET, JDBC, native C library, ODBC, Streaming API for large objects

InfluxDB vs TimeScaleDB - Data Structures

InfluxDB

```
name: cpu          name: cpu
tagKey            fieldKey fieldType
-----
cpu              iowait   float
hostname        used     float
```

TimescaleDB

```
Table "public.cpu"
  Column | Type | Collation | Nullable | Default | Storage | Stats target | Description
-----+-----+-----+-----+-----+-----+-----+-----
time    | timestamp without time zone | | not null | | plain | | |
cpu     | text | | | | extended | | |
hostname | text | | | | extended | | |
iowait  | double precision | | | | plain | | |
used    | double precision | | | | plain | | |

Child tables: _timescaledb_internal._hyper_2_3_chunk,
               _timescaledb_internal._hyper_2_4_chunk
```

InfluxDB vs TimeScaleDB - Data Query

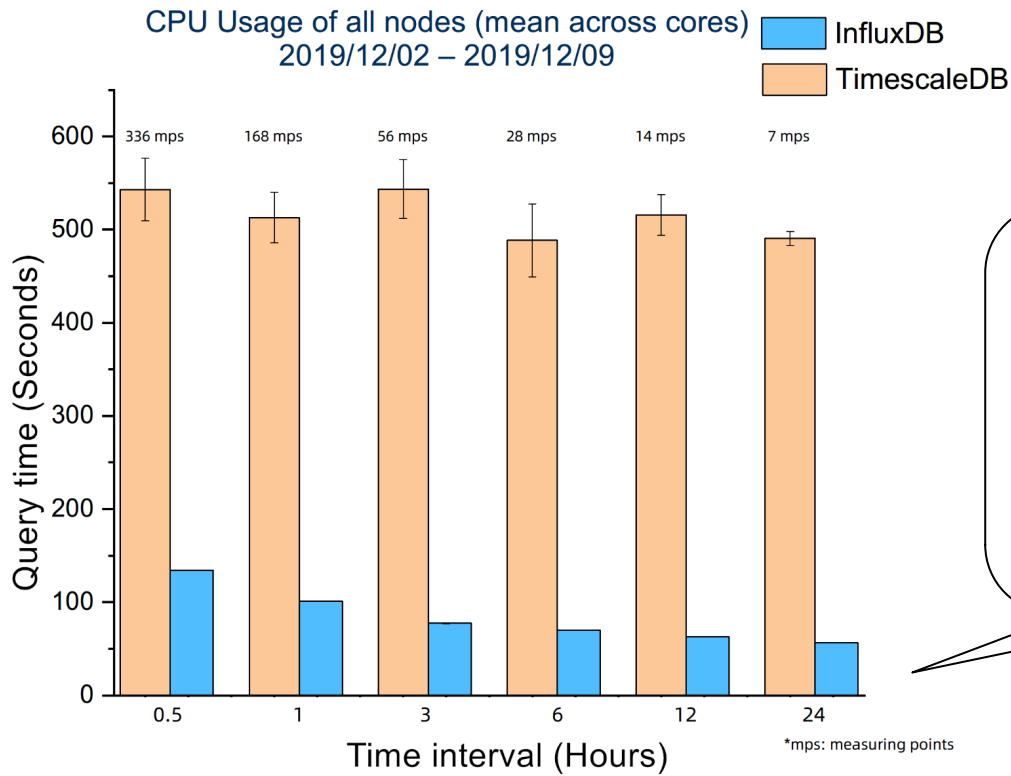
InfluxDB and InfluxQL

```
SELECT mean("used") AS value FROM "cpu"  
WHERE (((hostname='taurus15080') and (cpu='16' or cpu='17' or cpu='18' or cpu='19' or cpu='20' or cpu=  
or ((hostname='taurus15083') and (cpu='16' or cpu='17' or cpu='18' or cpu='19' or cpu='20' or cpu='21'  
.....  
or ((hostname='taurus15061') and (cpu='18' or cpu='19' or cpu='20' or cpu='21' or cpu='22' or cpu='23'  
AND ("time" >= 1575299111s AND "time" <= 1575759920s)  
GROUP BY time(3600s)
```

TimescaleDB and SQL

```
SELECT  
time_bucket('3600s','time') AS "time",  
avg(used) AS "used" FROM "cpu"  
WHERE  
(((hostname='taurus15080') and (cpu='16' or cpu='17' or cpu='18' or cpu='19' or cpu='20' or cpu='21'  
or ((hostname='taurus15083') and (cpu='16' or cpu='17' or cpu='18' or cpu='19' or cpu='20' or cpu='21'  
.....  
or ((hostname='taurus15061') and (cpu='18' or cpu='19' or cpu='20' or cpu='21' or cpu='22' or cpu='23'  
AND ("time" >= '2019-12-02T15:05:11Z' AND "time" <= '2019-12-07T23:05:20Z')
```

InfluxDB vs TimeScaleDB - CPU Usage System Query



Attention:
Performance differences currently not explainable. TimeScaleDB was installed with default settings.

InfluxDB vs TimeScaleDB - Storage

CPU Usage measurement
(table) for one week

Databases	Storage (Difference)
InfluxDB v1.8	6.25 GB
InfluxDB v0.9	34.8 GB
TimescaleDB v1.7	49 GB

