OSA CON 24

Exploring data analysis in time series databases

Aliaksandr Valialkin, CTO at VictoriaMetrics

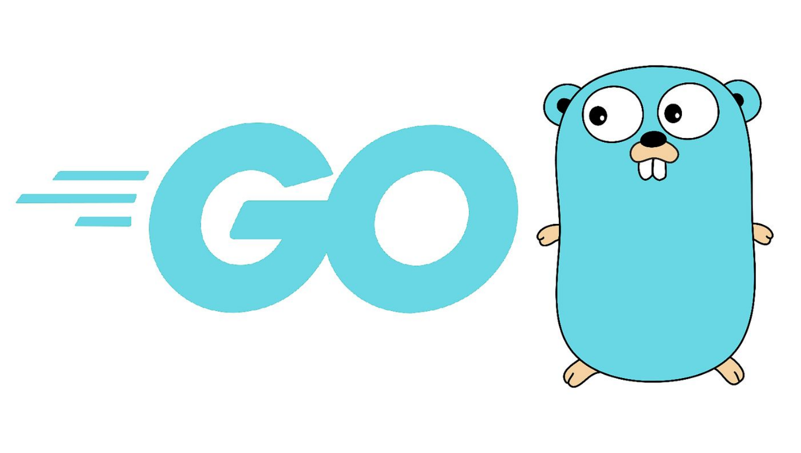
November 19-21, 2024

About me

- I’m software engineer

About me

- I’m software engineer
- I like writing fast programs in Go
  - Go is easy to use
  - Go is productive
  - Go programs can be fast

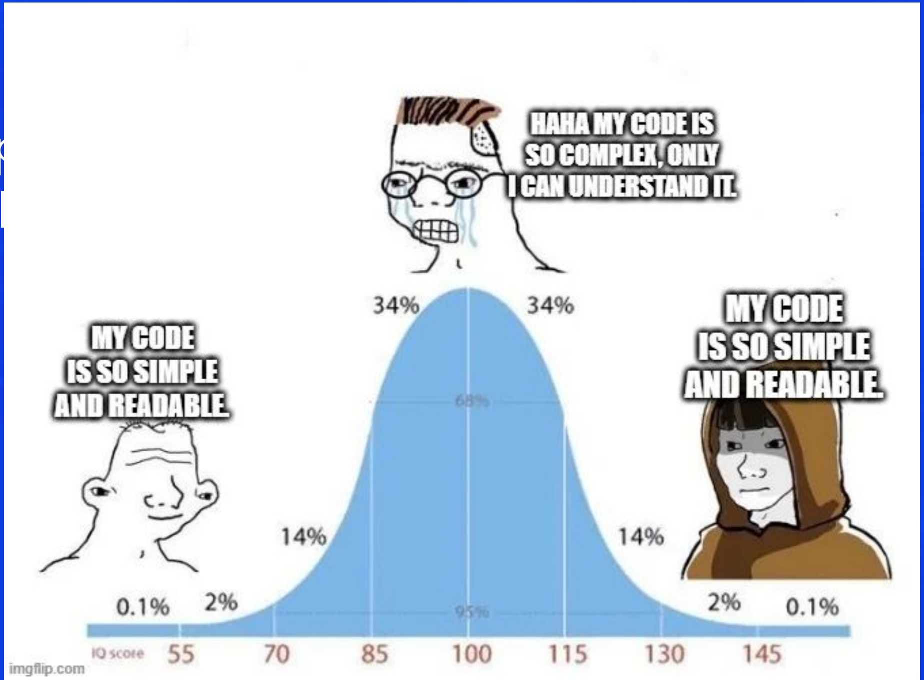


About me

- I’m software engineer
- I like writing fast programs in Go
- I hate over-complicated code with useless abstractions

About me

- I'm software engi
- I like writing fast p
- I hate over-compl


HAHA MY CODE IS SO COMPLEX, ONLY I CAN UNDERSTAND IT.
MY CODE IS SO SIMPLE AND READABLE.
MY CODE IS SO SIMPLE AND READABLE.
34%
34%
68%
14%
14%
2%
2%
0.1%
0.1%
95%
IQ score
55
70
85
100
115
130
145
imgflip.com

About me

- I’m software engineer
- I like writing fast programs in Go
- I hate over-complicated code with useless abstractions
- I hate over-complicated architectures with a lot of “magic”
  - Impossible to debug and recover when the “magic” breaks

About me

- I'm software engineer
- I like writing fast progra
- I hate over-complicated ons
- I hate over-complicated magic"
    - Impossible to debug anc

About me

- I’m software engineer
- I like writing fast programs in Go
- I hate over-complicated code with useless abstractions
- I hate over-complicated architectures with a lot of “magic”
- I’m the core developer of:
  - VictoriaMetrics - time-series database for metrics
  - VictoriaLogs - time-series database for logs

About me

- I’m software engineer
- I like writing fast programs in Go
- I hate over-complicated code with useless abstractions
- I hate over-complicated architectures with a lot of “magic”
- I’m the core developer of:
  - VictoriaMetrics - time-series database for metrics
  - VictoriaLogs - time-series database for logs
- I’m inspired by ClickHouse focus on performance

What is a time series?

What is a time series?

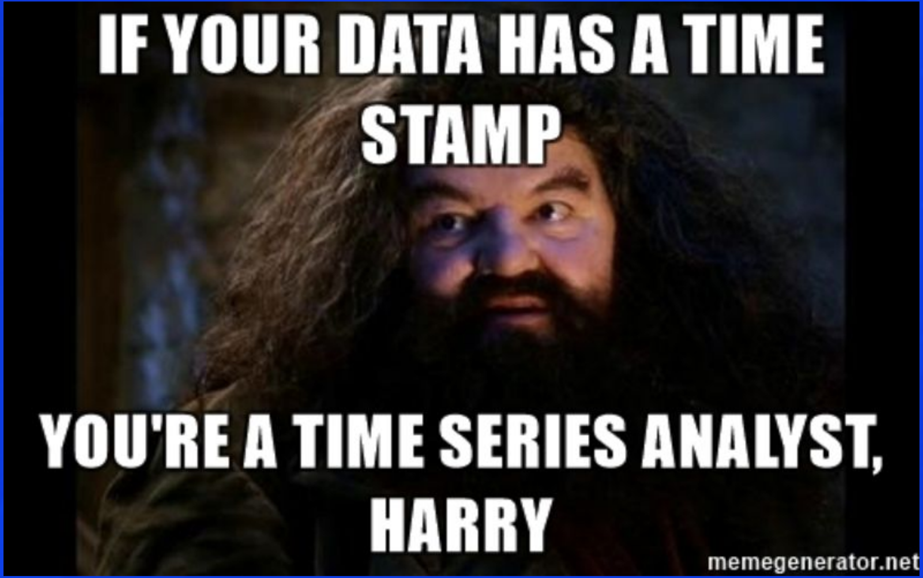
- A series of (timestamp; value) samples sorted by timestamp:

What is a time series?

- A series of (timestamp; value) samples sorted by timestamp:
    - (t1; v1), (t2; v2), … (tN; vN)

What is a time series?

- A series of (timestamp; value) samples sorted by timestamp:
  - (t1; v1), (t2; v2), … (tN; vN)


IF YOUR DATA HAS A TIME STAMP
YOU'RE A TIME SERIES ANALYST, HARRY
memegenerator.net

What is a time series?

- A series of (timestamp; value) samples sorted by timestamp:
  - (t1; v1), (t2; v2), … (tN; vN)
- Every time series usually has a name - temperature, memory_usage, etc.

What is a time series?

- A series of (timestamp; value) samples sorted by timestamp:
    - (t1; v1), (t2; v2), … (tN; vN)
- Every time series usually has a name - temperature, memory_usage, etc.
- Every time series may have arbitrary set of (label=value) labels:

What is a time series?

- A series of (timestamp; value) samples sorted by timestamp:
  - (t1; v1), (t2; v2), … (tN; vN)
- Every time series usually has a name - temperature, memory_usage, etc.
- Every time series may have arbitrary set of (label=value) labels:
  - temperature{city=”NY”,country=”US”}
  - memory_usage{host=”foo”,env=”prod”,az=”us-east”}

What is a time series?

- A series of (timestamp; value) samples sorted by timestamp:
  - (t1; v1), (t2; v2), … (tN; vN)
- Every time series usually has a name - temperature, memory_usage, etc.
- Every time series may have arbitrary set of (label=value) labels:
  - temperature{city="NY",country="US"}
  - memory_usage{host="foo",env="prod",az="us-east"}
- The set of (label=value) labels remain constant across all the (timestmap; value) samples, which belong to the same time series

What is a time series?

- A time series is uniquely identified by name{labels…}

What is a time series?

- A time series is uniquely identified by name{labels…}. The following are two different time series, since they differ by city label:
    - temperature{city=”NY”}
    - temperature{city=”SF”}

What is a time series?

- A time series is uniquely identified by name{labels…}. The following are two different time series, since they differ by city label:
  - temperature{city="NY"}
  - temperature{city="SF"}
- Example time series:
  - temperature{city="SF"}: (2024-11-10; 47°F), … (2024-11-19, 2024; 50°F)

What is a time series?

- A time series is uniquely identified by name{labels…}. The following are two different time series, since they differ by city label:
  - temperature{city=”NY”}
  - temperature{city=”SF”}
- Example time series:
  - temperature{city=”SF”}: (2024-11-10; 47°F), … (2024-11-19, 2024; 50°F)
  - cpu_usage_percent{host=”foo”,env=”prod”}: (10:20:30; 50%), … (23:34:59; 12%)

Time series value types

Time series value types

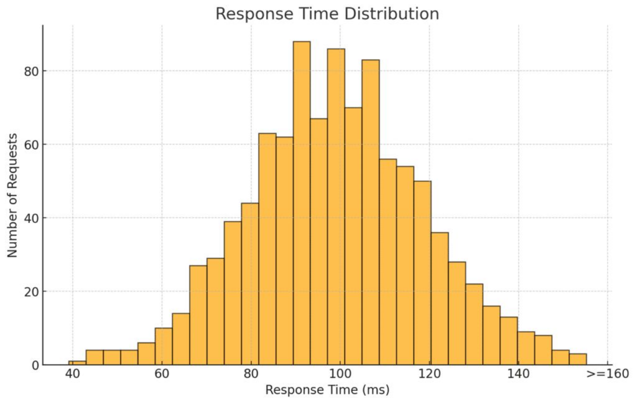
- Numeric values aka metrics or measurements:
  - temperature, cpu_usage, requests_total

Time series value types

- Numeric values aka metrics or measurements:
  - temperature, cpu_usage, requests_total
- Non-numeric values

Time series value types

- Numeric values aka metrics or measurements:
  - temperature, cpu_usage, requests_total
- Non-numeric values
  - Histograms - every value contains a set of histogram buckets:
    - (t1; buckets1), … (tN; bucketsN)



Time series value types

- Numeric values aka metrics or measurements:
    - temperature, cpu_usage, requests_total
- Non-numeric values
    - Histograms - every value contains a set of histogram buckets:
        - (t1; buckets1), … (tN; bucketsN)
    - Plaintext logs - every value contains plaintext log message:
        - (t1; msg1), … (tN; msgN)

Time series value types

- Numeric values aka metrics or measurements:
  - temperature, cpu_usage, requests_total
- Non-numeric values
  - Histograms - every value contains a set of histogram buckets:
    - (t1; buckets1), … (tN; bucketsN)
  - Plaintext logs - every value contains plaintext log message:
    - (t1; msg1), … (tN; msgN)
  - Structured logs aka events - every value contains a set of (field=value) fields:
    - (t1; fields1), … (tN; fieldsN)

Time series value types

- Numeric values aka metrics or measurements:
  - temperature, cpu_usage, requests_total
- Non-numeric values
  - Histograms - every value contains a set of histogram buckets:
    - (t1; buckets1), … (tN; bucketsN)
  - Plaintext logs - every value contains plaintext log message:
    - (t1; msg1), … (tN; msgN)
  - Structured logs aka events - every value contains a set of (field=value) fields:
    - (t1; fields1), … (tN; fieldsN)
  - Wide events - every value contains hundreds of (field=value) fields

My knowledge about TV series
My knowledge about time series

What is a time series database?

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
  - Trillions of rows (raw samples)

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
  - Trillions of rows (raw samples)
  - Hundreds of terabytes of data

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
- It usually provides data ingestion APIs optimized for typical time series data

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
- It usually provides data ingestion APIs optimized for typical time series data
  - OpenTelemetry for logs, traces and metrics
  - Prometheus text exposition format
  - Graphite metrics format
  - Influx line protocol
  - Elasticsearch for logs

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
- It usually provides data ingestion APIs optimized for typical time series data
- It usually provides query language optimized for typical queries over time series data

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
- It usually provides data ingestion APIs optimized for typical time series data
- It usually provides query language optimized for typical queries over time series data
  - PromQL and MetricsQL for metrics
  - LogsQL for logs

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
- It usually provides data ingestion APIs optimized for typical time series data
- It usually provides query language optimized for typical queries over time series data
- Time series databases can specialize for particular value types:

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
- It usually provides data ingestion APIs optimized for typical time series data
- It usually provides query language optimized for typical queries over time series data
- Time series databases can specialize for particular value types:
  - Metrics - Prometheus, Thanos, VictoriaMetrics

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
- It usually provides data ingestion APIs optimized for typical time series data
- It usually provides query language optimized for typical queries over time series data
- Time series databases can specialize for particular value types:
    - Metrics - Prometheus, Thanos, VictoriaMetrics
    - Logs - Grafana Loki, VictoriaLogs

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
- It usually provides data ingestion APIs optimized for typical time series data
- It usually provides query language optimized for typical queries over time series data
- Time series databases can specialize for particular value types:
  - Metrics - Prometheus, Thanos, VictoriaMetrics
  - Logs - Grafana Loki, VictoriaLogs
  - Events - VictoriaLogs

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
- It usually provides data ingestion APIs optimized for typical time series data
- It usually provides query language optimized for typical queries over time series data
- Time series databases can specialize for particular value types:
  - Metrics - Prometheus, Thanos, VictoriaMetrics
  - Logs - Grafana Loki, VictoriaLogs
  - Events - VictoriaLogs
  - Traces - Grafana Tempo

What is a time series database?

- A database optimized for efficient storing and querying of large volumes of time series data
- It usually provides data ingestion APIs optimized for typical time series data
- It usually provides query language optimized for typical queries over time series data
- Time series databases can specialize for particular value types:
  - Metrics - Prometheus, Thanos, VictoriaMetrics
  - Logs - Grafana Loki, VictoriaLogs
  - Events - VictoriaLogs
  - Traces - Grafana Tempo
  - Mixed - InfluxDB, ClickHouse

What about using relational databases such as Postgresql or MySQL for time series data?

What about using relational databases such as Postgresql or MySQL for time series data?

- OK if the number of samples doesn’t exceed a few billions

What about using relational databases such as Postgresql or MySQL for time series data?

- OK if the number of samples doesn’t exceed a few billions
- Not OK for bigger number of samples:

What about using relational databases such as Postgresql or MySQL for time series data?

- OK if the number of samples doesn’t exceed a few billions
- Not OK for bigger number of samples:
  - High disk space usage because of inefficient data compression

What about using relational databases such as Postgresql or MySQL for time series data?

- OK if the number of samples doesn’t exceed a few billions
- Not OK for bigger number of samples:
    - High disk space usage because of inefficient data compression
        - Traditional RDBMS: 1TB of logs occupies 2TB of disk space

What about using relational databases such as Postgresql or MySQL for time series data?

- OK if the number of samples doesn’t exceed a few billions
- Not OK for bigger number of samples:
  - High disk space usage because of inefficient data compression
    - Traditional RDBMS: 1TB of logs occupies 2TB of disk space
    - Database optimized for logs: 1TB of logs occupies 20GB of disk space

What about using relational databases such as Postgresql or MySQL for time series data?

- OK if the number of samples doesn’t exceed a few billions
- Not OK for bigger number of samples:
  - High disk space usage because of inefficient data compression
  - High disk read IO usage because of inefficient storage layout

What about using relational databases such as Postgresql or MySQL for time series data?

- OK if the number of samples doesn’t exceed a few billions
- Not OK for bigger number of samples:
    - High disk space usage because of inefficient data compression
    - High disk read IO usage because of inefficient storage layout
        - Traditional RDBMS: makes a million of reads from random locations on disk for fetching a million of samples for the typical query, while reading 10GB of data from disk

What about using relational databases such as Postgresql or MySQL for time series data?

- OK if the number of samples doesn't exceed a few billions
- Not OK for bigger number of samples:
  - High disk space usage because of inefficient data compression
  - High disk read IO usage because of inefficient storage layout
    - Traditional RDBMS: makes a million of reads from random locations on disk for fetching a million of samples for the typical query, while reading 10GB of data from disk
    - Database optimized for time series data: makes a few sequential reads for fetching a million of samples for the typical query, while reading 1MB of data from disk

What about using relational databases such as Postgresql or MySQL for time series data?

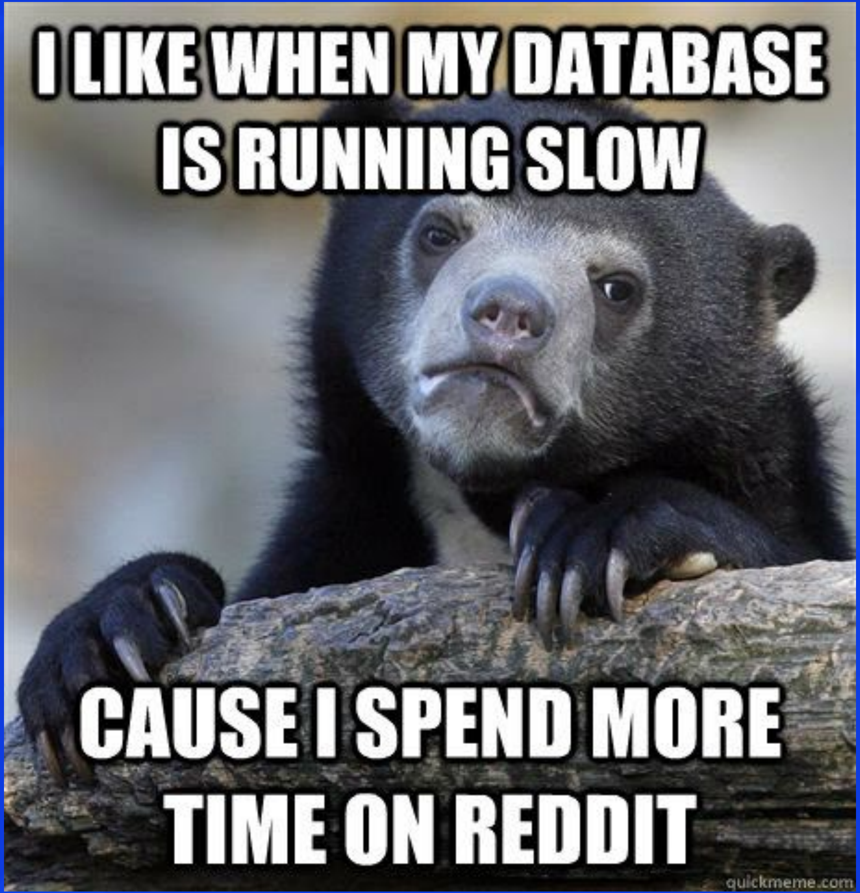
- OK if the number of samples doesn’t exceed a few billions
- Not OK for bigger number of samples:
    - High disk space usage because of inefficient data compression
    - High disk read IO usage because of inefficient storage layout
    - High RAM usage because of inefficient indexes

What about using relational databases such as Postgresql or MySQL for time series data?

- OK if the number of samples doesn’t exceed a few billions
- Not OK for bigger number of samples:
    - High disk space usage because of inefficient data compression
    - High disk read IO usage because of inefficient storage layout
    - High RAM usage because of inefficient indexes
        - Traditional RDBMS: occupies all the available RAM and makes a million of random reads from disk per query, because the index for 10 trillions of rows doesn’t fit RAM

What about using relational databases such as Postgresql or MySQL for time series data?

- OK if the number of samples doesn't exceed a few billions
- Not OK for bigger number of samples:
  - High disk space usage because of inefficient data compression
  - High disk read IO usage because of inefficient storage layout
  - High RAM usage because of inefficient indexes
    - Traditional RDBMS: occupies all the available RAM and makes a million of random reads from disk per query, because the index for 10 trillions of rows doesn't fit RAM
    - Time series database: occupies a few GiB of RAM for sparse index over 10 trillions of rows


I LIKE WHEN MY DATABASE IS RUNNING SLOW
CAUSE I SPEND MORE TIME ON REDDIT
quickmeme.com

Challenges with Kubernetes monitoring

Challenges with Kubernetes monitoring

- Typical Kubernetes clusters contain thousands of containers

Challenges with Kubernetes monitoring

- Typical Kubernetes clusters contain thousands of containers
- Every container generates many metrics, logs and events

Challenges with Kubernetes monitoring

- Typical Kubernetes clusters contain thousands of containers
- Every container generates many metrics, logs and events
- The summary number of samples for metrics, logs and events exceeds tens of billions per day

Challenges with Kubernetes monitoring

- Typical Kubernetes clusters contain thousands of containers
- Every container generates many metrics, logs and events
- The summary number of samples for metrics, logs and events exceeds tens of billions per day
  - A thousand of containers, which expose 500 metrics each, and these metrics are stored into database per every 10 seconds, generate 1000*500*(24*3600/10)=4.32 billions of samples per day

Challenges with Kubernetes monitoring

- Typical Kubernetes clusters contain thousands of containers
- Every container generates many metrics, logs and events
- The summary number of samples for metrics, logs and events exceeds tens of billions per day
  - A thousand of containers, which expose 500 metrics each, and these metrics are stored into database per every 10 seconds, generate 1000*500*(24*3600/10)=4.32 billions of samples per day
  - A thousand of containers, which generate 100 events (logs) per second each, result in 1000*10*24*3600=8.64 billions of events per day

Challenges with Kubernetes monitoring

- All these samples must be **efficiently** stored, so they occupy the smallest possible amounts of disk space

Challenges with Kubernetes monitoring

- All these samples must be efficiently stored, so they occupy the smallest possible amounts of disk space
- All these samples must be efficiently queried with a query language optimized for typical monitoring tasks: alerting, analyzing dashboards with graphs for the requested metrics/logs/events over the selected time range

Kubernetes monitoring: which database to use?

Kubernetes monitoring: which database to use?

- Relational databases such as Postgresql or MySQL?

Kubernetes monitoring: which database to use?

- Relational databases such as Postgresql or MySQL? No, since they aren’t optimized for trillions of samples

Kubernetes monitoring: which database to use?

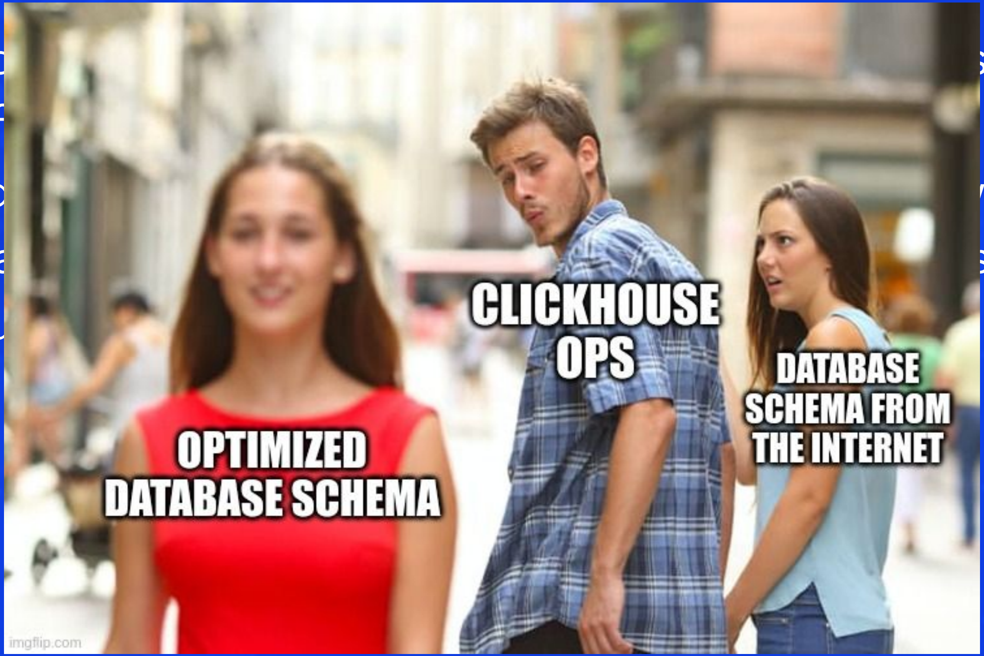
- Relational databases such as Postgresql or MySQL? No, since they aren’t optimized for trillions of samples
- Analytical databases such as ClickHouse?

Kubernetes monitoring: which database to use?

- Relational databases such as Postgresql or MySQL? No, since they aren't optimized for trillions of samples
- Analytical databases such as ClickHouse? Yes if you know how to create the best database schema for Kubernetes monitoring use cases in order to get the maximum performance and storage efficiency

Kubernetes monitoring: which database to use?

- Relational c since they aren't optimized f
- Analytical c how to create the best databa ses in order to get the maximu



Kubernetes monitoring: which database to use?

- Relational databases such as Postgresql or MySQL? No, since they aren't optimized for trillions of samples
- Analytical databases such as ClickHouse? Yes if you know how to create the best database schema for Kubernetes monitoring use cases in order to get the maximum performance and storage efficiency
- Specialized databases such as VictoriaMetrics and VictoriaLogs?

Kubernetes monitoring: which database to use?

- Relational databases such as Postgresql or MySQL? No, since they aren't optimized for trillions of samples
- Analytical databases such as ClickHouse? Yes if you know how to create the best database schema for Kubernetes monitoring use cases in order to get the maximum performance and storage efficiency
- Specialized databases such as VictoriaMetrics and VictoriaLogs? Yes, since they work great for particular cases (metrics, logs, events and wide events) without any configuration and tuning.

Kubernetes monitoring: which database to use?

- Relational databases such as Postgresql or MySQL? No, since they aren't optimized for trillions of samples
- Analytical databases such as ClickHouse? Yes if you know how to create the best database schema for Kubernetes monitoring use cases in order to get the maximum performance and storage efficiency
- Specialized databases such as VictoriaMetrics and VictoriaLogs? Yes, since they work great for particular cases (metrics, logs, events and wide events) without any configuration and tuning. They may work slower than optimized ClickHouse database schema though

Which query language to use?

Which query language to use?

- SQL (with analytical extensions from ClickHouse)

Which query language to use?

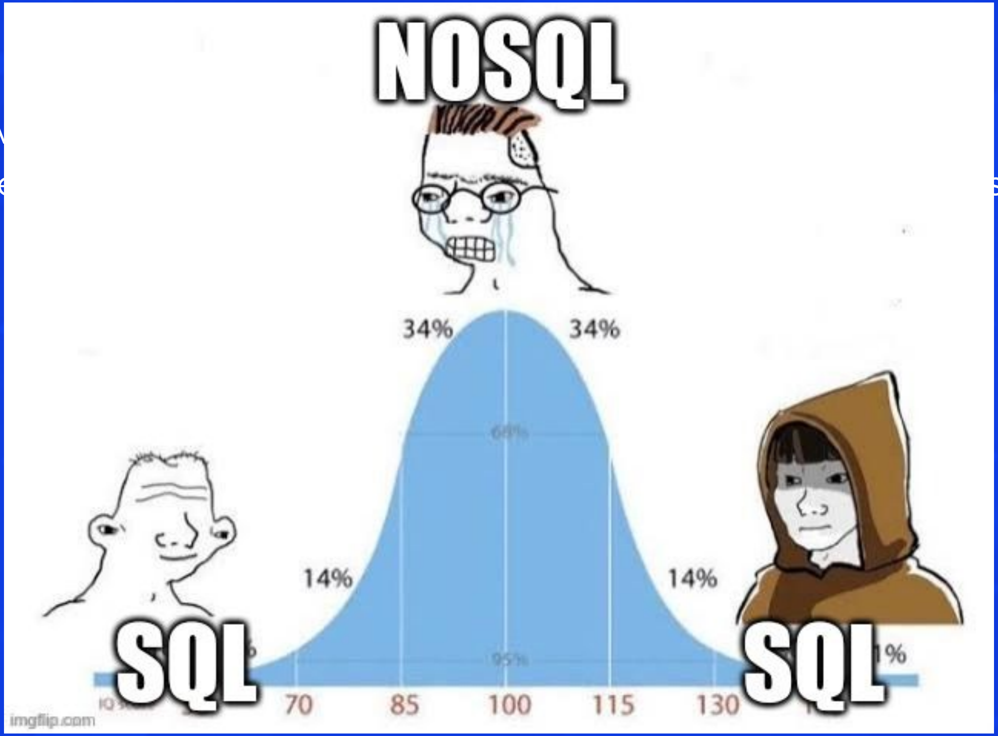
- SQL (with analytical extensions from ClickHouse)
  - It is expressive enough to cover the majority of monitoring use cases

Which query language to use?

- SQL (with analytical extensions from ClickHouse)
  - It is expressive enough to cover the majority of monitoring use cases
  - It covers cases, which cannot be covered by specialized query languages

Which query language to use?

- SQL (with anal
  - It is expressiv
  - It covers case s



Which query language to use?

- SQL (with analytical extensions from ClickHouse)
  - It is expressive enough to cover the majority of monitoring use cases
  - It covers cases, which cannot be covered by specialized query languages
  - The resulting queries for typical monitoring use cases are usually too long and too complex to manage by an average user (DevOps, SRE)

Which query language to use?

- SQL (with analytical extensions from ClickHouse)
  - It is expressive enough to cover the majority of monitoring use cases
  - It covers cases, which cannot be covered by specialized query languages
  - The resulting queries for typical monitoring use cases are usually too long and too complex to manage by an average user (DevOps, SRE)
- Specialized query language: PromQL, MetricsQL, LogsQL

Which query language to use?

- SQL (with analytical extensions from ClickHouse)
  - It is expressive enough to cover the majority of monitoring use cases
  - It covers cases, which cannot be covered by specialized query languages
  - The resulting queries for typical monitoring use cases are usually too long and too complex to manage by an average user (DevOps, SRE)
- Specialized query language: PromQL, MetricsQL, LogsQL
  - Easier to write queries for typical monitoring use cases

Which query language to use?

- SQL (with analytical extensions from ClickHouse)
  - It is expressive enough to cover the majority of monitoring use cases
  - It covers cases, which cannot be covered by specialized query languages
  - The resulting queries for typical monitoring use cases are usually too long and too complex to manage by an average user (DevOps, SRE)
- Specialized query language: PromQL, MetricsQL, LogsQL
  - Easier to write queries for typical monitoring use cases
  - You need to learn yet another query language

Which query language to use?

- SQL (with analytical extensions from ClickHouse)
  - It is expressive enough to cover the majority of monitoring use cases
  - It covers cases, which cannot be covered by specialized query languages
  - The resulting queries for typical monitoring use cases are usually too long and too complex to manage by an average user (DevOps, SRE)
- Specialized query language: PromQL, MetricsQL, LogsQL
  - Easier to write queries for typical monitoring use cases
  - You need to learn yet another query language
  - Some queries are harder or impossible to write

Typical data analysis tasks for time series data

- Select time series with the given set of labels

SQL

```
SELECT
  id
FROM
  series
WHERE
  hasAll(labels, ['label1=value1', ... 'labelN=valueN'])
```

Typical data analysis tasks for time series data

- Select time series with the given set of labels

PromQL

```
{label1="value1", ... labelN="valueN"}
```

Typical data analysis tasks for time series data

- Select time series with the given set of labels
- Select samples for the selected time series on the given time range

SQL

```sql
SELECT
  series_id
  timestamp,
  value
FROM samples
WHERE
  series_id IN (<query_from_the_previous_slide>)
  AND timestamp > $start AND timestamp <= $end
```

Typical data analysis tasks for time series data

- Select time series with the given set of labels
- Select samples for the selected time series on the given time range

PromQL

```
{label1="value1", ... labelN="valueN"}[$__range]

/api/v1/query?time=$end&query=...
```

Typical data analysis tasks for time series data

- Select time series with the given set of labels
- Select samples for the selected time series on the given time range
- Apply some aggregations over the selected time series

SQL

```
SELECT
  series_id,
  avg(value) AS avg_value
FROM (<query_from_the_prevous_slide>)
GROUP BY
  series_id
```

Typical data analysis tasks for time series data

- Select time series with the given set of labels
- Select samples for the selected time series on the given time range
- Apply some aggregations over the selected time series

PromQL

```
avg_over_time({label1="value1", ... labelN="valueN"}[$__range])
```

Typical data analysis tasks for time series data

- Select time series with the given set of labels
- Select samples for the selected time series on the given time range
- Apply some aggregations over the selected time series
- Apply some filters on the calculated aggregates

SQL

```
SELECT
  series_id,
  avg_value
FROM (<query_from_the_prevous_slide>)
WHERE avg_value > $threshold
```

Typical data analysis tasks for time series data

- Select time series with the given set of labels
- Select samples for the selected time series on the given time range
- Apply some aggregations over the selected time series
- Apply some filters on the calculated aggregates

PromQL

```
avg_over_time(
  {label1="value1", ... labelN="valueN"}[$__range]
) > $threshold
```

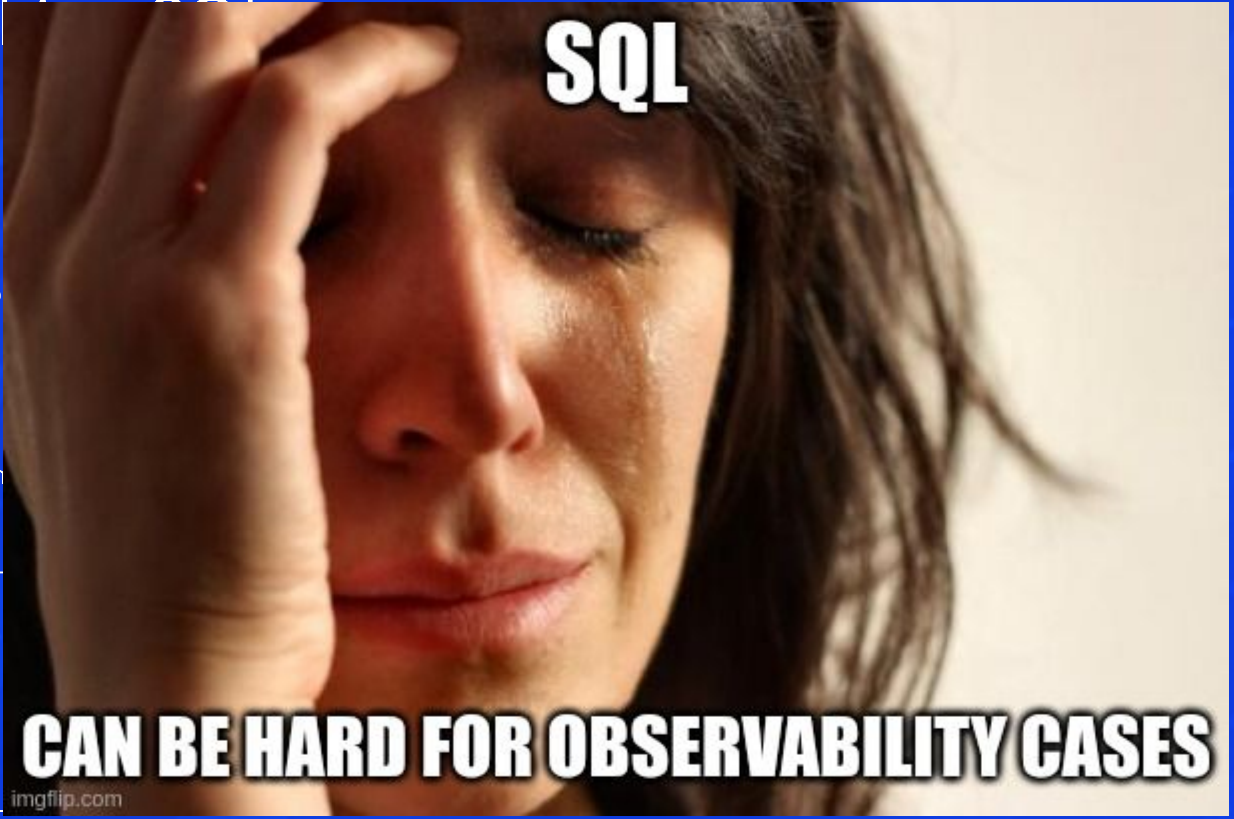
The resulting SQL query

The resulting SQL query

```
SELECT
  series_id,
  avg_value
FROM (
  SELECT
    series_id,
    avg(value) AS avg_value
  FROM (
    SELECT
      series_id,
      value
    FROM samples
    WHERE
      series_id IN (
        SELECT id FROM series WHERE hasAll(labels, ['label1=value1', ... 'labelN=valueN'])
      )
      AND timestamp > $start AND timestamp <= $end
  )
  GROUP BY series_id
)
WHERE avg_value > $threshold
```

The resul

```
SELECT
  series_id,
  avg_value
FROM (
  SELECT
    series_id,
    avg(value)
  FROM (
    SELECT
      series_id
      value
    FROM san
    WHERE
      series_id
        SELECT
      )
      AND tim
  )
  GROUP BY
)
WHERE avg_
```



Example queries: SQL vs PromQL

Select temperature in NY city on the given time range

Example queries: SQL vs PromQL

Select temperature in NY city on the given time range

SQL

```
SELECT
  date_trunc('$__step', timestamp) AS timestamp_truncated,
  last_value(value) AS temperature
FROM
  metrics
WHERE
  series_id IN (
    SELECT id FROM series
    WHERE __name__ = 'temperature' AND city = 'NY'
  )
  AND timestamp > now() - $__range AND timestamp < now()
GROUP BY
  timestamp_truncated
ORDER BY
  timestamp_truncated
```

Example queries: SQL vs PromQL

Select temperature in NY city on the given time range

SQL

```
SELECT
  date_trunc('$__step', timestamp) AS timestamp_truncated,
  last_value(value) AS temperature
FROM
  metrics
WHERE
  series_id IN (
    SELECT id FROM series
    WHERE __name__ = 'temperature' AND city = 'NY'
  )
  AND timestamp > now() - $__range AND timestamp < now()
GROUP BY
  timestamp_truncated
ORDER BY
  timestamp_truncated
```

PromQL or MetricsQL

```
temperature{city="NY"}
```

Example queries: SQL vs LogsQL

Select top 5 applications with the biggest number of error logs, which do not contain “broken pipe” phrase during the last day

Example queries: SQL vs LogsQL

Select top 5 applications with the biggest number of error logs, which do not contain “broken pipe” phrase during the last day

SQL

```
SELECT
  app,
  count(*) hits
FROM
  logs
WHERE
  timestamp > now() - 1 day AND timestamp <= now()
  AND level = 'error'
  AND positionUTF8(_msg, 'broken pipe') = 0
GROUP BY
  app
ORDER BY
  hits DESC
LIMIT 5
```

Example queries: SQL vs LogsQL

Select top 5 applications with the biggest number of error logs, which do not contain “broken pipe” phrase during the last day

SQL

```
SELECT
  app,
  count(*) hits
FROM
  logs
WHERE
  timestamp > now() - 1 day AND timestamp <= now()
  AND level = 'error'
  AND positionUTF8(_msg, 'broken pipe') = 0
GROUP BY
  app
ORDER BY
  hits DESC
LIMIT 5
```

LogsQL

```
_time:1d level:=error -"broken pipe"
    | top 5 by (app)
```

Example queries: SQL vs LogsQL

Select the number of apps with more than 10 error logs per each step on the graph

Example queries: SQL vs LogsQL

Select the number of apps with more than 10 error logs per each step on the graph

SQL

```
SELECT
  timestamp_trucated,
  count()
FROM (
  SELECT
    date_trunc('$__step', timestamp) AS timestamp_truncated,
    app,
    count(*) hits
  FROM logs
  WHERE
    timestamp >= now() - $__range AND timestamp < now()
    AND level = 'error'
  GROUP BY timestamp_truncated, app
  HAVING hits > 10
)
GROUP BY timestamp_truncated
ORDER BY timestamp_truncated
```

Example queries: SQL vs LogsQL

Select the number of apps with more than 10 error logs per each step on the graph

SQL

```
SELECT
  timestamp_trucated,
  count()
FROM (
  SELECT
    date_trunc('$__step', timestamp) AS timestamp_truncated,
    app,
    count(*) hits
  FROM logs
  WHERE
    timestamp >= now() - $__range AND timestamp < now()
    AND level = 'error'
  GROUP BY timestamp_truncated, app
  HAVING hits > 10
)
GROUP BY timestamp_truncated
ORDER BY timestamp_truncated
```

LogsQL

```
level:=error
    | stats by (app) count() as hits
    | hits:>10
    | count()
```

Why PromQL and LogsQL is easier than SQL for graph building in Grafana?

- Because Prometheus, VictoriaMetrics and VictoriaLogs provide specialized HTTP-based APIs for such queries

Why PromQL and LogsQL is easier than SQL for graph building in Grafana?

- Because Prometheus, VictoriaMetrics and VictoriaLogs provide specialized HTTP-based APIs for such queries
- These APIs accept **start**, **end** and **step** args additionally to the query itself:
  - /api/v1/query_range?start=…&end=…&step=…&query=…
  - /select/logsql/stats_query_range?start=…&end=…&step=…&query=…

Why PromQL and LogsQL is easier than SQL for graph building in Grafana?

- Because Prometheus, VictoriaMetrics and VictoriaLogs provide specialized HTTP-based APIs for such queries
- These APIs accept **start**, **end** and **step** args additionally to the query itself:
  - /api/v1/query_range?start=...&end=...&step=...&query=...
  - /select/logsql/stats_query_range?start=...&end=...&step=...&query=...
- The query is automatically executed per every **step** on the **[start … end]** time range

Why PromQL and LogsQL is easier than SQL for graph building in Grafana?

- Because Prometheus, VictoriaMetrics and VictoriaLogs provide specialized HTTP-based APIs for such queries
- These APIs accept **start**, **end** and **step** args additionally to the query itself:
  - /api/v1/query_range?start=...&end=...&step=...&query=...
  - /select/logsql/stats_query_range?start=...&end=...&step=...&query=...
- The query is automatically executed per every **step** on the **[start … end]** time range
- The query results are grouped and returned per each step at the following timestamps: **start, start+step, start+2*step, … end**

Conclusions

Conclusions

- Typical time series data volumes do not fit relational databases

Conclusions

- Typical time series data volumes do not fit relational databases
- Prefer specialized databases for storing and querying time series data

Conclusions

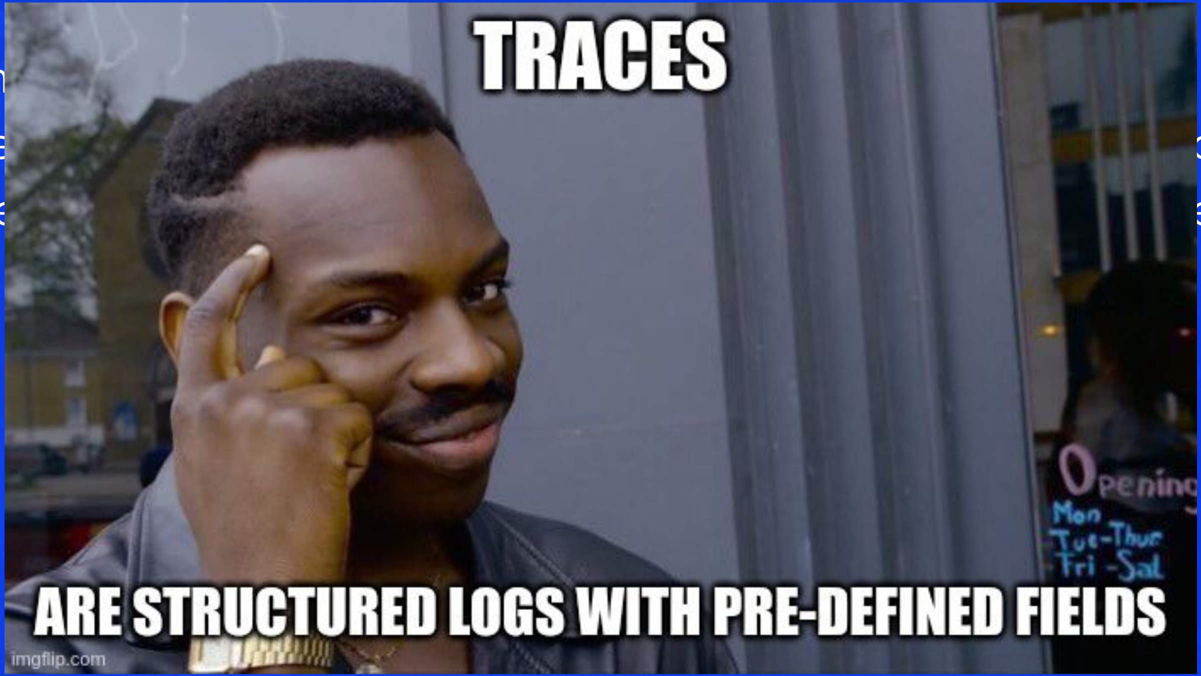
- Typical time series data volumes do not fit relational databases
- Prefer specialized databases for storing and querying time series data
- Time series data includes: metrics, logs, events and wide events

Conclusions

- Typical time series data volumes do not fit relational databases
- Prefer specialized databases for storing and querying time series data
- Time series data includes: metrics, logs, events and wide events
  - What about traces?

Conclusions

- Typical tin
- Prefer spe data
- Time serie events



Conclusions

- Typical time series data volumes do not fit relational databases
- Prefer specialized databases for storing and querying time series data
- Time series data includes: metrics, logs, events and wide events
- Prefer specialized databases for particular time series data types

RDBMS
TSDB
DEVOPS DECIDING WHICH DATABASE TYPE TO USE FOR TIME SERIES DATA
JAKE-CLARK.TUMBLR
imgflip.com

Questions?

- ClickHouse docs - https://clickhouse.com/docs
- VictoriaMetrics docs - https://docs.victoriametrics.com/
- VictoriaLogs docs - https://docs.victoriametrics.com/victorialogs/
- LogsQL docs - https://docs.victoriametrics.com/victorialogs/logsql/