Time Series Databases

With demonstrations using InfluxDB

Time Series

- A series of data points given in time order
- A point can have multiple values (tuple). Commonly measurements, samples, or readings at regular intervals
- A point can optionally have multiple tags or labels (metadata) for identifying or filtering a contained time series within a larger time series
- Interest lies in the series as a whole

Time Series Analysis

- Large subject matter
- Numerous uses. Some examples:
  - Forecasting and prediction
  - Trends
  - Classification
  - Function approximation

Time Series Databases

- https://db-engines.com/en/ranking/time+series+dbms
- Wide column databases
- Traditional RDBMS with extensions

General Characteristics of Time Series Databases

- Data retention
- Data downsampling and consolidation
- Analytic processing
- Common use-case is for monitoring

General Deployment of Monitoring Stack

- Collectors
- Back-end storage engine (Time Series DB) and analytic engine
- Visualization system
- Notification system

RRDTool

- Origins in MRTG in 1999. RRDTool is still popular.
- Each interval has a Primary Data Point (PDP)
- Consolidation functions map multiple PDPs into a Consolidation Data Point (CDP)
- Configurable number of CDPs are store
- Retention interval = interval/PDP * PDPs/CDP * CPDs

OpenTSDB

- Runs on Hadoop and HBase
- Highly scalable
- *Would consider OpenTSDB for a sizable production system but wanted to go simpler for my project*

Graphite

- Three major components (Python):
  - Carbon (Twisted) — Listener
  - Whisper — Time Series DB similar to RRDTool
  - Graphite Web (Django) — Graphing
- Graphite use either a simple String-base protocol or a binary protocol influenced by Python

InfluxDB

- Developed as open source by Errplane[*] in 2013 [* later to become InfluxData]
- No external dependencies
- Supports many plugins to accept data using common listeners:
  - UDP
  - Collectd
  - Graphite (Carbon)
  - OpenTSDB

InfluxDB Time Series Identification

- Measurement
- Tag Set
- Retention Policy

InfluxDB protocol

- HTTP(S)-based
- POSTed parameters can specify database (required), retention policy, user, password, precision of type stamp.
- Measurement data is in the HTTP body and is text based:

<M>[,<T_K>=<T_V>] <F_K>=<F_V>[,<F_K>=<F_V>] [T]

InfluxDB Data Types

<M>[,<T_K>=<T_V>] <F_K>=<F_V>[,<F_K>=<F_V>] [T]

- Measurement, Tag keys and values, and Field keys are Strings
- Field values can be floats (default for numbers), integers [both 64-bit], strings or boolean
- Timestamp is nanosecond Unix time

InfluxDB Indexes

<M>[,<T_K>=<T_V>] <F_K>=<F_V>[,<F_K>=<F_V>] [T]

- Timestamp is indexed
- Tag set is indexed (inverted in-memory index)
  - Tag keys should be sorted for insert
  - Cardinality of Tag set affects memory usage
- **Field set is not indexed**

InfluxDB Continuous Queries

- Run automatically and periodically on real time data. Must GROUP BY time

- Stores results in a specified measurement [SELECT … INTO …], possibly with a different retention policy

- Common use-case to downsample or transform measurements:

```
CREATE CONTINUOUS QUERY "cq_basic_rp" ON "transportation"

BEGIN

SELECT mean("passengers") INTO "transportation"."three_weeks"."average_passengers"
FROM "bus_data" GROUP BY time(1h)

END
```

InfluxDB Functions

- Aggregations, Transformations, Predictions
- Some still have not been implemented
- https://docs.influxdata.com/influxdb/v1.2/query_language/functions/

InfluxDB Demonstrations