SQL history development

Gennaro Tortone – INFN Napoli

Midas Workshop 2023

MIDAS history system

- MIDAS **mlogger** can record slow control data and/or periodic events to disk
- Multiple backend supported:
  - MIDAS history
  - MYSQL
  - ODBC
  - SQLITE
  - PostgreSQL (new)
- Users can view history data using <u>MIDAS history web page</u>
  - user selects a time window
  - data are requested by web browser through JSON-RPC method from JavaScript
 - received data is plotted on custom web plot canvas

MIDAS history system

- – Potential issue:
  - what happens if user selects a large timeperiod (e.g. several months) and there are hundreds of megabytes of data ?
    - – user's browser crashes (!) due to max memory usage limit
- – Possible solution:
  - plot canvas maximum number of points (x axes) is less than or equal to monitor horizontal resolution (1K, 2K, 4K, …)
  - **downsampling** of whole data set can help to reduce N points to few thousand (e.g 4000) preserving plot "shape"
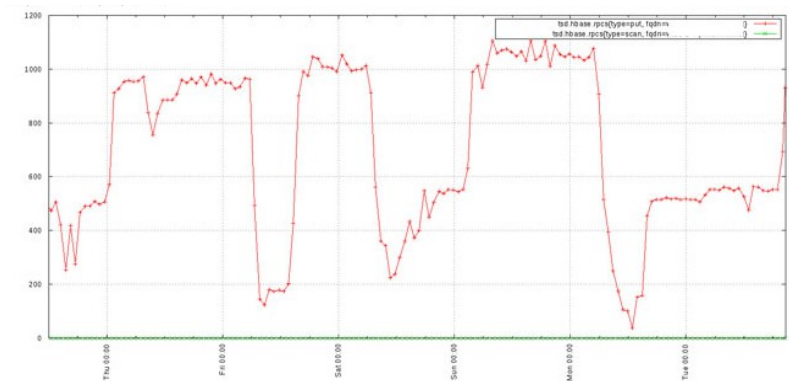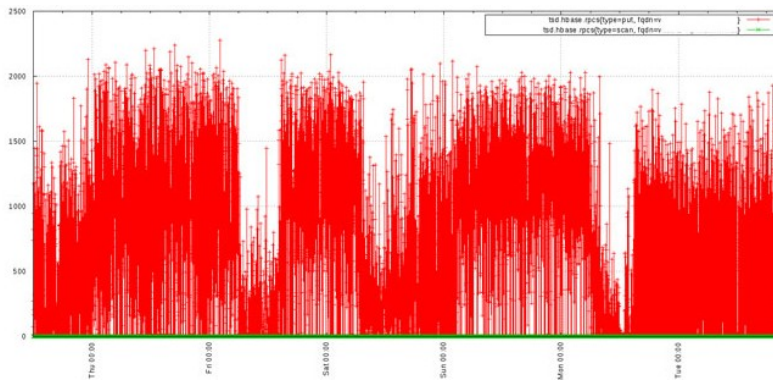
Downsampling

- Algorithms for time series data
  - **binning**: aggregate multiple points in a single points (bin) using a function (e.g. arithmetic mean) [statistical aggregation]
  - **LTTB** (Largest-Triangle-Three-Buckets): reduces the number of **visually** redundant data points, resulting in smaller file sizes and faster rendering of plots [visual aggregation]
  - … **Simple Moving Average**, **Automatic Smoothing for Attention Prioritization**, etc.
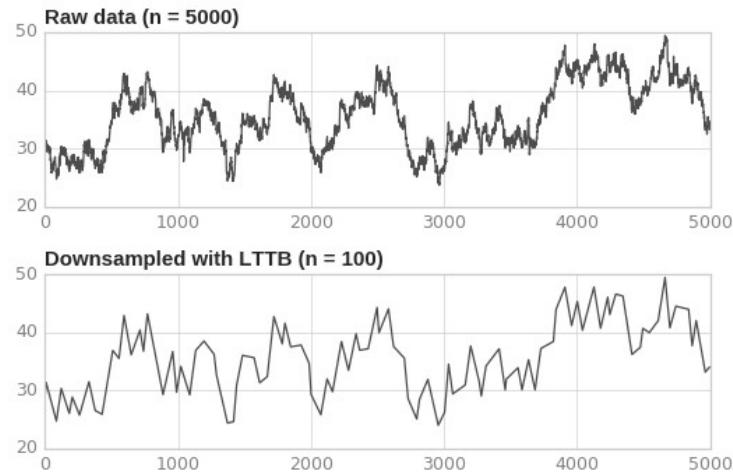
Downsampling

- Where apply downsampling in MIDAS

  - web browser (user side)

    - does not solve the memory allocation issue in case of large dataset, raw data must be fetched and allocated in JavaScript

  - mhttpd (server side)

    - data allocation happens on remote server

    - a custom/generic downsampling algorithm can be developed in order to use it on different database backend (also on simple history data files…)

    - in case of concurrent queries with large number of records the memory allocation issue can happen on mhttpd...

Downsampling

- Where apply downsampling in MIDAS (continue)

  - SQL database backend

    - move database backend to TSDB (**Time Series DataBase**)
    - downsampling can be requested in SQL query
    - memory allocation is not an issue due to software design of TSDB backend
    - downsampling algorithms supported by TSDB are a predefined set
    - solution not suitable for simple MIDAS history files (no backend)

Time Series DataBase

- TSDB is a (non-relational - **NoSQL**) database management system that is optimized to store, process, and analyze time-series data
- Use cases:
    - sensors on industrial equipments
    - smart devices
    - IT monitoring systems
    - stock market trades
- TSDB offers:
    - high data ingestion rate
    - low query latency
    - optimization of storage costs (data compression)

Time Series DataBase

- different product exists…
  - standalone DB server software (e.g. InfluxDB, OpenTSDB)
  - extension of well-known RDBMS (e.g TimescaleDB / PostgreSQL)

Time Series DataBase

- Examples of queries on TSDB

```sql
SELECT symbol, first(price,time), last(price, time)
FROM stocks_real_time srt
WHERE time > now() - INTERVAL '4 days'
GROUP BY symbol
ORDER BY symbol
LIMIT 10;
```

```sql
SELECT
    time_bucket('1 day', "time") AS day,
    symbol,
    max(price) AS high,
    first(price, time) AS open,
    last(price, time) AS close,
    min(price) AS low
FROM stocks_real_time srt
```

Hypertables
chunk_time_interval = "1 day"

Normal table

time	value
2021-01-02 00:00:00	36
2021-01-02 06:00:00	5
2021-01-02 23:00:00	29
2021-01-03 00:00:00	17
2021-01-03 06:00:00	8
2021-01-03 23:00:00	6
2021-01-04 00:00:00	41
2021-01-04 06:00:00	14
2021-01-04 23:00:00	5

Hypertable

time	value
Chunk ID 1	
2021-01-02 00:00:00	36
2021-01-02 06:00:00	5
2021-01-02 23:00:00	29
Chunk ID 2	
2021-01-03 00:00:00	17
2021-01-03 06:00:00	8
2021-01-03 23:00:00	6
Chunk ID 3	
2021-01-04 00:00:00	41
2021-01-04 06:00:00	14
2021-01-04 23:00:00	5

Conclusions

- MIDAS team is involved in history data downsampling

- future release of MIDAS will provide downsampling mechanism on 'mhttpd' side

- different strategies/solutions exist, users will benefit of these new features and will suggest further ways to go !

Thanks !