Time Series Databases for Mere Mortals

Michael Zelenetz

- I assume no prior experience
- I am not an expert
- I have no conflicts of interest or financial stake in Timescale

About Me

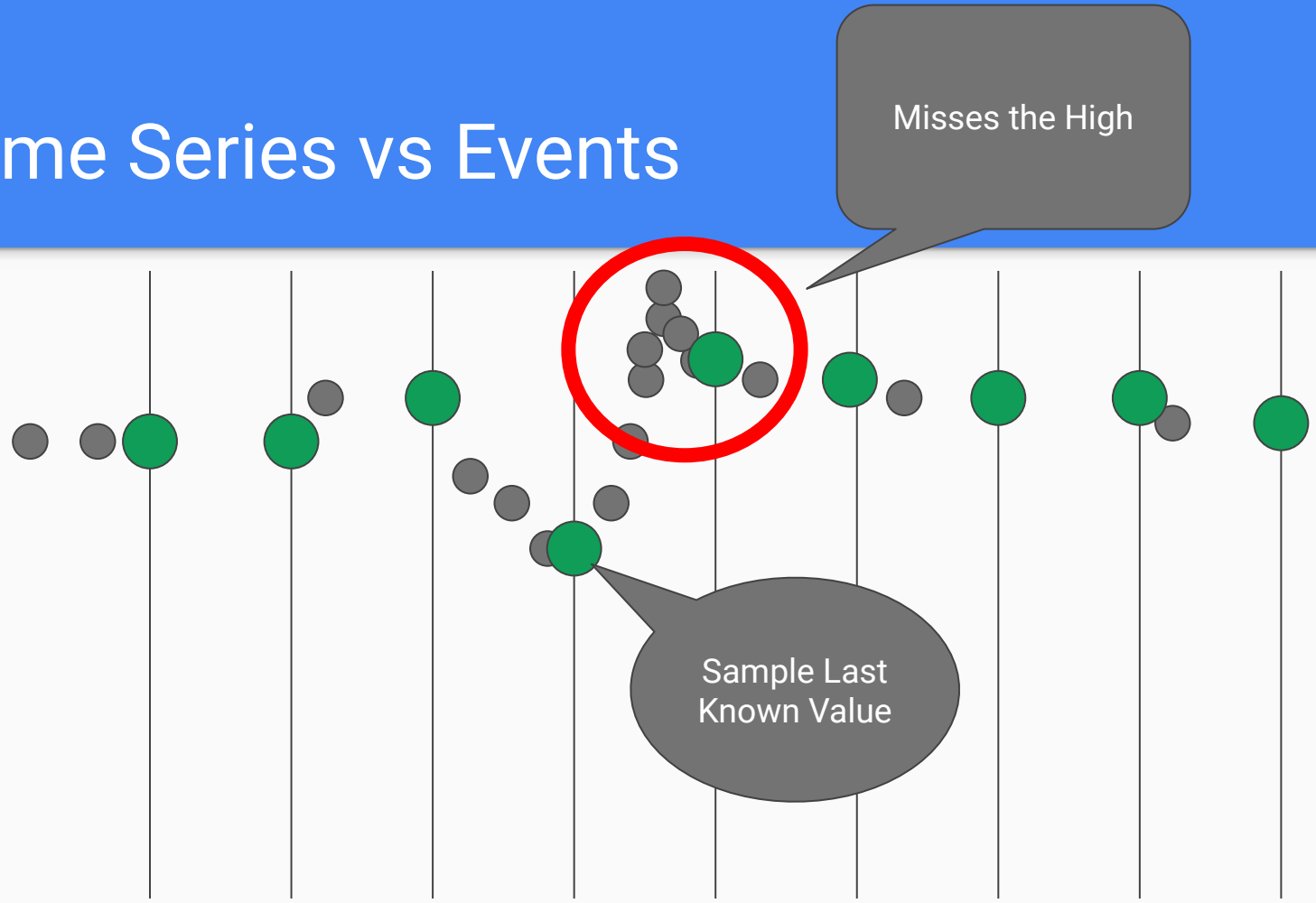
- Trading Data @ Peak6 Capital Management
  - Primarily Trading Options
- Adjunct Faculty at Yeshiva University
- Previously Healthcare
- NYHACKR attendee since 2016

What is a Time Series?

- Sampled at consistent time interval
  - (second, minute, hour, day, …)
- Generally append only
- Approximates a continuous value
- Table structure: (entity_id, timestamp), value

Weather Station	Timestamp	Temp
Central Park	12:00 PM	72.2
Albany	12:00PM	68.2
Central Park	1:00 PM	74.5
Albany	1:00 PM	72.1

Time Series vs Events



How did I get involved in Timescale?

- Thousands of Time Series
- Dozens of Terabytes
- Supporting Research and Applications
- Supports user defined TS

Timescale DB

Some Background on Timescale

- Open Source Database
- Postgres Extension
  - Lives alongside relational data
- Full SQL support
- Hybrid row-columnar storage
- Compression
- Time series specific functions

https://github.com/timescale/timescaledb

1000x

Faster Query Performance

(Compared to internal legacy benchmark query)

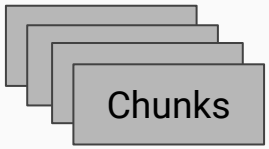
95%

Disk Space Savings

Building Blocks

Fundamental Building Blocks in Timescale

SQL Version	Timescale DB
View	Continuous Aggregates
Table	Hypertables



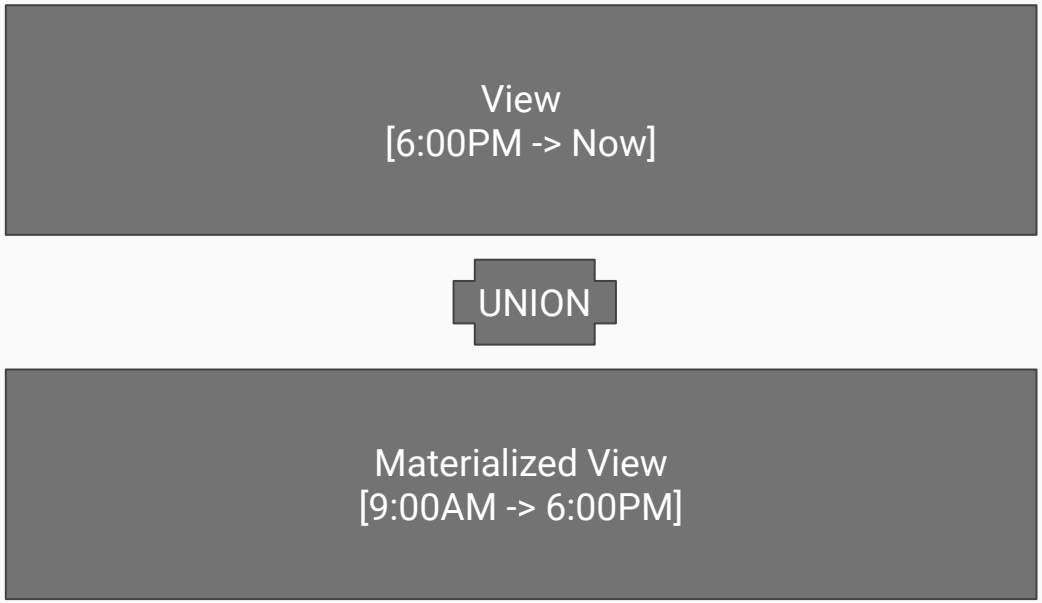
Hypertables

- Vanilla Postgres Table to Hypertable
  - `select create_hypertable('stock_prices', by_range('time'));`
- TS Handles chunking/partitioning
- Join with other PG Tables

Continuous Aggregates

- View
  - Saved Query
- Materialized View
  - Needs refresh

Continuous Aggregate



Internal Scheduler

Timescale handles scheduling

- Continuous Aggregates
- Compression
- Retention

Adding Policies is just SQL

```
SELECT add_compression_policy('stock_prices',
          compress_after => INTERVAL '1d');
```

Compression

Row to Column

Time	Symbol	Price
12:00:01	AAPL	210
12:00:02	AAPL	211
12:00:01	NVDA	129
12:00:02	NVDA	130

Time	Symbol	Price
[12:00:01,12:00:02]	AAPL	[210,211]
[12:00:01,12:00:02]	NVDA	[129,130]

Compression Algos

Delta Encoding: How much has the value changed from the prior value (int)

Run Length Encoding: very repetitive data (11,11,11,11,11,11 –> {6:11})

Dictionary Compression: replace commonly repeated words/labels with int

Gorilla (float): Save the first float, XOR subsequent values and save that

id	version	name	description
0	1	COMPRESSION_ALGORITHM_NONE	no compression
1	1	COMPRESSION_ALGORITHM_ARRAY	array
2	1	COMPRESSION_ALGORITHM_DICTIONARY	dictionary
3	1	COMPRESSION_ALGORITHM_GORILLA	gorilla
4	1	COMPRESSION_ALGORITHM_DELTADELTA	deltadelta

Built in Time Functions

- time_bucket
- first
- last

Timescale Toolkit

- Candlestick_agg
- Gap filling
- Time Weighted Calcs

Enough Talk
Now Code

Familiar easy to use SQL interface
Batteries included
Great documentations
Scales nicely (so far)

Hypertables
Continuous Aggregates
Policies
    Compression
    Retention

Michael Zelenetz

 @mzelenetz

 michael.zelenetz@gmail.com