Final Presentation

Carnegie Mellon University

Metrics Forecasting

Dhruv Arya // Kai Franz // Neville Chima // 15-799 // Spring 2022

METRICS FORECASTING

1. Predict future metrics based only on historical data — time series forecasting
2. Forecasts for metrics like dead tuple percentage — can be used to intelligently schedule table vacuuming.
3. Different from MB2 — does not rely on an additional workload forecasting step.
4. Currently tracked metrics: dead tuple count and table growth.

Scope

Adds a new module metrics_forecaster

Inputs:

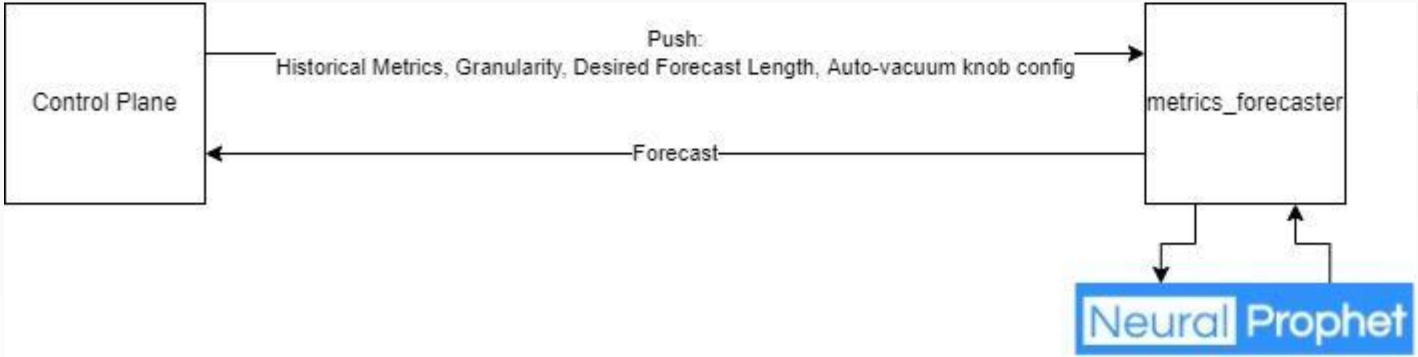
1. Historical metrics data (timeseries, one set per table)
2. Required granularity of predictions (e.g. 5s, 20s)
3. Required forecast length (how far into the future?)

Outputs:

1. Predicted metric values based on the historical data

Currently supported metrics: table_size, dead tuple count

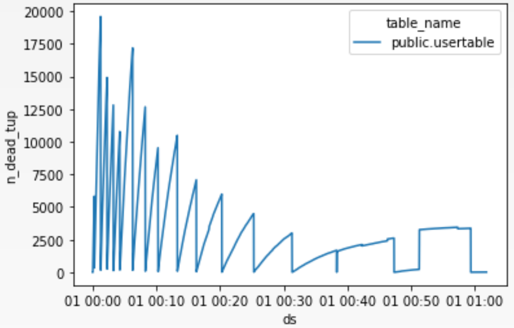
Architectural Design

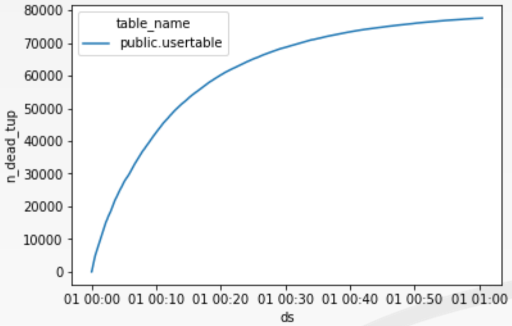


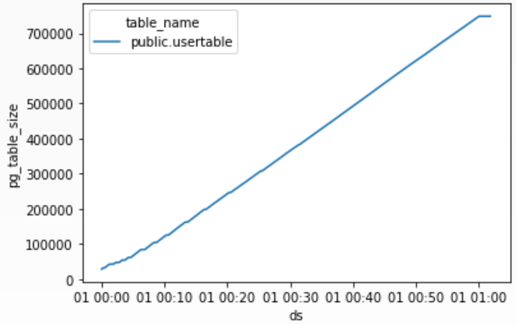
Data Collected

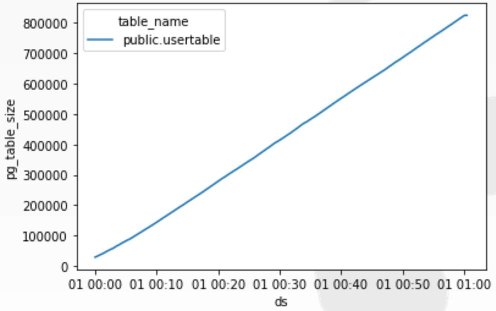
Benchmark	Benchmark Time (s)	Scale Factor	Auto-vacuum settings	Data Collected	Additional config
SmallBank	3600	50/100	on/off	table size, n_dead_tuples	
TPCC	3600, 7200	50/100	on/off	table size, n_dead_tuples	
YCSB	3600, 7200	50/100	on/off	table size, n_dead_tuples	DeleteRecord weight set to 25
TATP	3600	50/100	on/off	table size, n_dead_tuples	

AutoVacuum vs No AutoVacuum (YCSB)









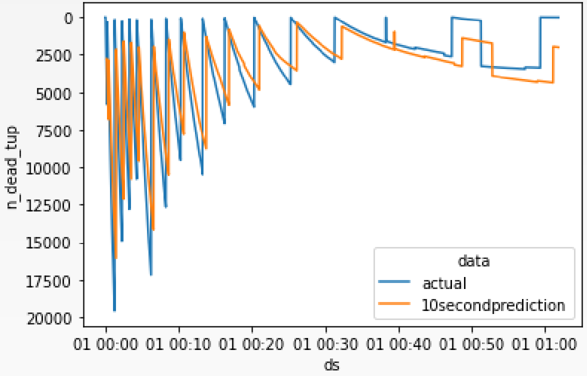
3600 second run with a sampling rate of once per second; scale factor = 50

YCSB (Prediction 10s into the future)

Autovac on

Autovac off

Dead tuples



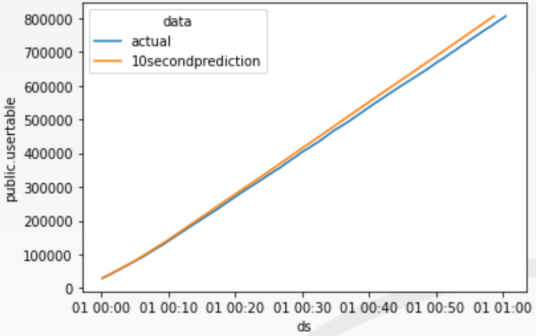
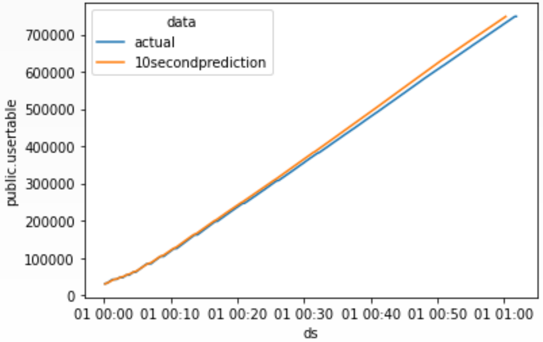
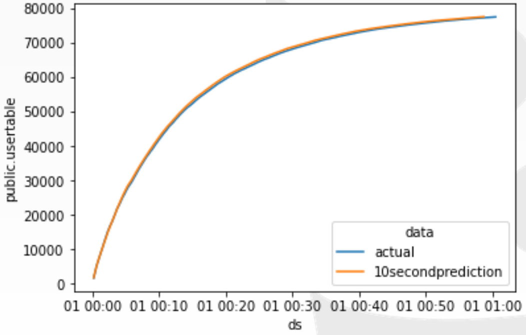


Table
growth



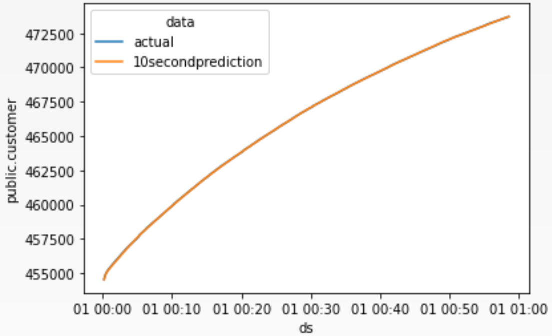


TPC-C Table Growth Forecasting

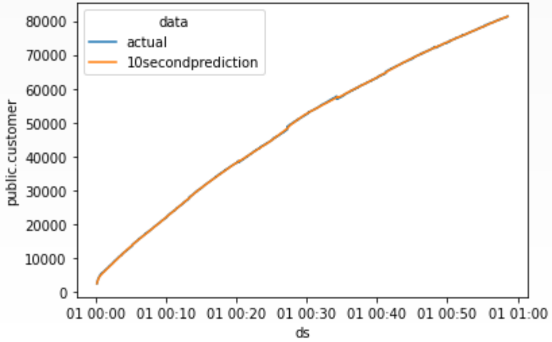
Autovacuum ON

`customer`

Table size



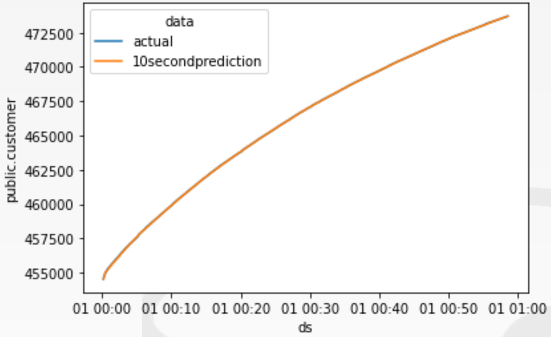
Dead tuples



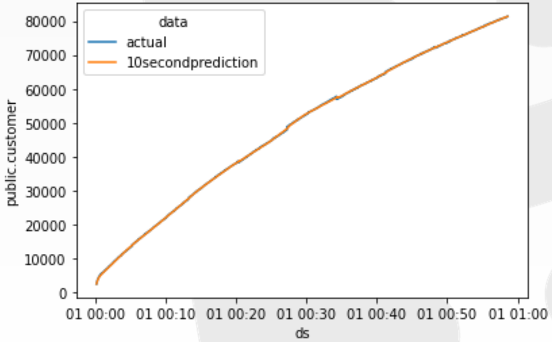
Autovacuum OFF

`customer`

Table size



Dead tuples



Results

	YCSB	TPCC_customer
	MAE	MAE
Table Growth Vacuum ON	59.3	11.0
Table Growth Vacuum OFF	39.9	10.8
Dead Tuple Count Vacuum ON	79.2	84.9
Dead Tuple Count Vacuum OFF	64.4	49.4

Testing Plan

1. Store metric traces for all benchmarks in Benchbase with deletes
2. Test for training time on each benchmark:
   a. With normal trace with a 50:50 train-test split
   b. Artificially inflated trace just for test time
3. Test for correctness (MAE, RMSE):
   a. With a 50:50 split
   b. For data binned at granularity levels: 1 minute, 1 hour, 1 day, 1 week
4. Continuously log correctness metrics values per horizon evaluated

Trade-offs and Potential Problems

1. Neural prophet simply creates a linear projection for the future when predicting far into the future

2. Neural prophet takes around 13s to fit on 3600 samples of data — need to test for very large samples

3. Design does not take transferability into account. Needs to be trained from scratch on each database.

Future Work

1. Real/Longer workload traces (week, month)

2. Address the cold start problem (map to previously seen workloads?)

1. Account for different auto-vacuum configurations

2. Test other models: ARIMA, FBProphet, LSTMs

QUESTIONS