Forecasting++ Update

Jia Qi Dong, Yingjie Ling, Wan Shen Lim
Overview

- An overview of the development status of their project as related to the goals discussed in the initial proposal.

Proposal had three components:

A. Transaction-aware forecasting ← NeuralProphet, Markov chains
B. Forecasting parameters ← statistical, deep
C. Forecasting database state ← dropped

And proposed the following evaluation:

- 75%: have at least one component set up ← we have A and B
- 100%: have a baseline pipeline that handles numeric schemas ← WIP
- 125%: beat the baseline pipeline ← WIP
Deviations

- Due to time constraints, we have dropped forecasting future database state in favor of focusing on generating the query workload.

- If we forecast query parameters well, we can still get the future database state (by replaying the query workload). The reverse is not true.
Code coverage / testing

- A measurement of the current code coverage of the tests for your implementation.

- The current testing plan is to run our queries on PostgreSQL to synthesize a complete query log for our forecasted queries (“forecast log”).
- We will then compare the forecast log with the future queries in the query log (“future log”).
- For both the forecast log and the future log, we will (1) restore the initial state from a dump and (2) run pgreplay. Then compare various execution metrics and PostgreSQL statistics to see how they differ.
- Unfortunately, we are not able to robustly test for runtime beyond exposing various tqdm progress bars in the ML components.
General architecture

Query Logs → normalization → PostgreSQL CSVLOGs → splitting → Split CSVLOGs → preprocessing → Parquet

Parquet → streaming update → Forecast Metadata

- Query Template Encoder
- Historical Query Parameters
- Think Time Sketch
- BEGIN Arrival History
Forecasting query templates
Parameter Forecasting Workflow

Query Arrival Rate

- Standardize & Aggregate
  - Query 1 Agg
    - Compute Quantile
      - Param 1
        - Forecasting Model
    - Compute Quantile
      - Param 2
        - Forecasting Model
  - Standardize & Aggregate
    - Query 2 Agg
      - Compute Quantile
        - Param 1
          - Forecasting Model
      - Compute Quantile
        - Param 2
          - Forecasting Model
LSTM VS. DistFit

DELETE FROM new_order WHERE NO_O_ID = $1 AND NO_D_ID = $2 AND NO_W_ID = $3
Can Capture Various Trends

DELETE FROM new_order WHERE NO_O_ID = $1 AND NO_D_ID = $2 AND NO_W_ID = $3
Challenges

- **DistFit**
  - Cannot fit a distribution for data it has never seen.

- **One model for all**
  - Difficult to generalize; might require a lot of training data.

- **One model for one template**
  - Embed position information into the quantile data; middle ground.

- **One model for one parameter**
  - Storage/computation overhead scales with the number of parameters.
Future Work

- **DL model**
  - Online training
  - Confidence interval
  - Multivariate parameter prediction
  - String prediction

- **Dataset**
  - Currently TPCC
  - Test on real workload

- **Transaction-aware parameter forecasting**
  - Different parameter distribution for the same template in different sessions
  - Constraint on parameter value for different templates in the same session

Q1: SELECT * FROM warehouse WHERE \textit{w-id}=x;

Q2: SELECT * FROM district WHERE \textit{w-id}=x \text{ AND } \textit{d-id}=y;

Interference model predicts two $w_{id}$ should be equal