Self-Driving Database Management Systems

Yushan Qin

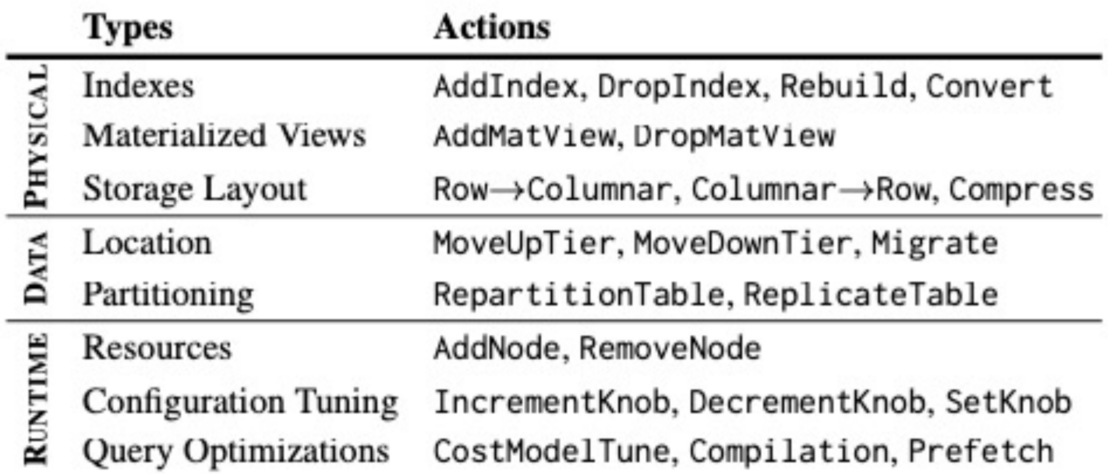
UW-Madison

Self-driving Database

- No human intervention in making decisions about changes to database
- Autonomic configure, tune, and optimize itself

- Database Design
- Data Placement
- Query Optimization

- Knob Configuration
- Back-up & Recovery
- Provisioning



	Types	Actions
PHYSICAL	Indexes	AddIndex, DropIndex, Rebuild, Convert
	Materialized Views	AddMatView, DropMatView
	Storage Layout	Row→Columnar, Columnar→Row, Compress
DATA	Location	MoveUpTier, MoveDownTier, Migrate
	Partitioning	RepartitionTable, ReplicateTable
RUNTIME	Resources	AddNode, RemoveNode
	Configuration Tuning	IncrementKnob, DecrementKnob, SetKnob
	Query Optimizations	CostModelTune, Compilation, Prefetch

What is Peloton

- A self-driving SQL database management system
- Integrating deep learning framework for workload forecasting and action deployment
- Uses a variant of multi-version concurrency control that interleaves OLTP transactions and actions without blocking OLAP queries
- Uses an in-memory storage manager with lock-free data structures and flexible layouts that allows for fast execution of HTAP workloads
- Uses LLVM to do query compilation and code generation.

Peloton-Monitor

Workload Monitor

- collecting relevant information about all running workloads. In addition to SQL queries, it records data manipulation quantities, resource utilization, and other related metrics
- The stream is also periodically punctuated with (1) DBMS/OS telemetry data and (2) the begin/end events for optimization actions.

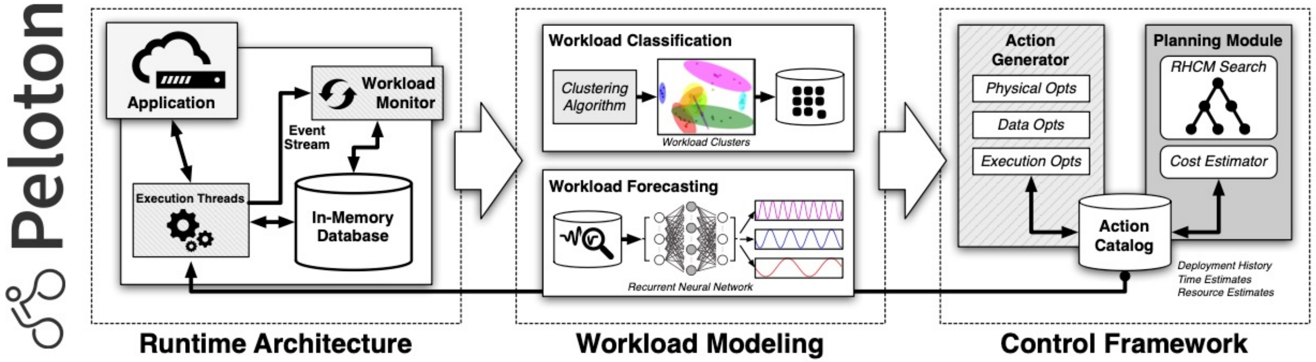


Figure 1: Peloton Self-Driving Architecture – An overview of the forecasting and runtime workflow of Peloton's self-driving components.

Peloton-Modelling

Clustering

- Cluster OLTP query by DBSCAN
- Retrain when error rate > threshold

Forecasting

- Integrated classified query on time-series
- Use Multiple RNNs to predict the arrival rate of queries for each workload cluster
- Also constructs similar models for the other DBMS/OS metrics in the event stream

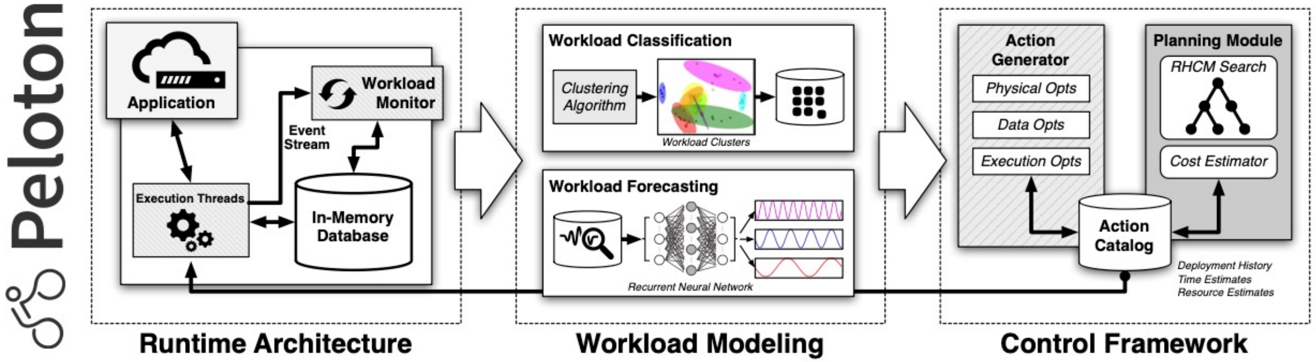
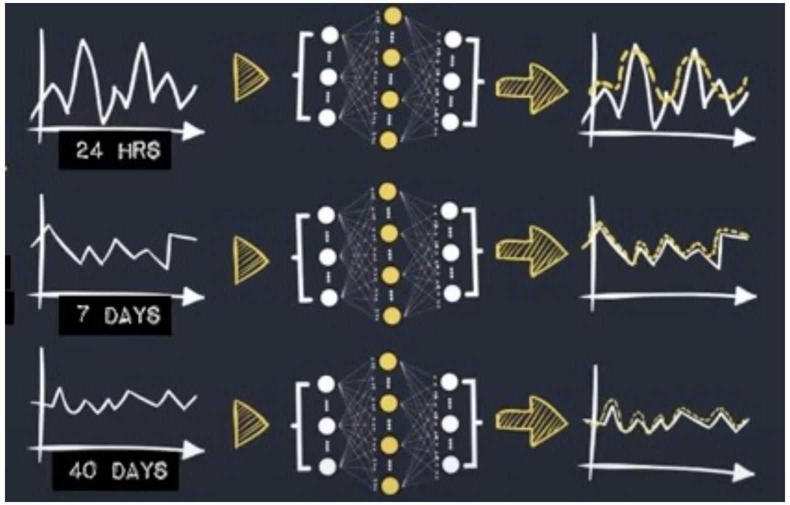


Figure 1: Peloton Self-Driving Architecture – An overview of the forecasting and runtime workflow of Peloton's self-driving components.



Peloton-Optimization

Action Generator

- Store many actions with corresponding consequences that may improve performance, select action based on forecasts, the current database configuration, and objective function

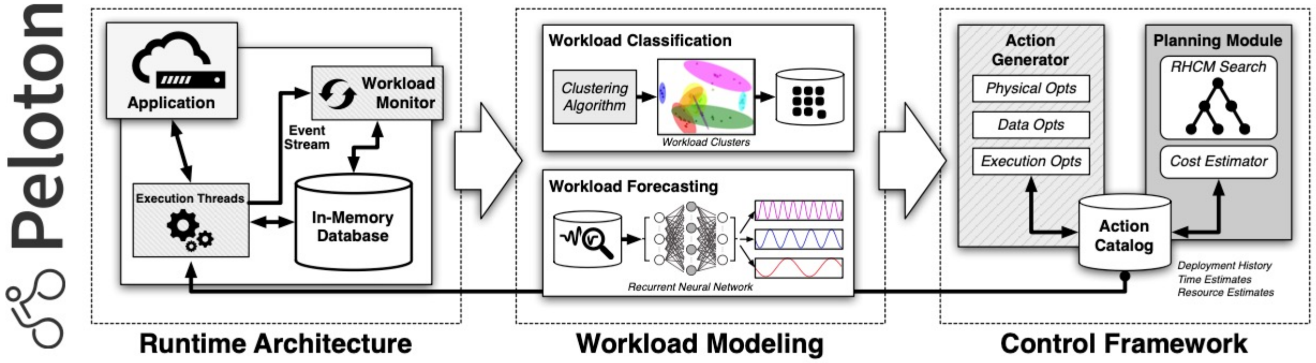
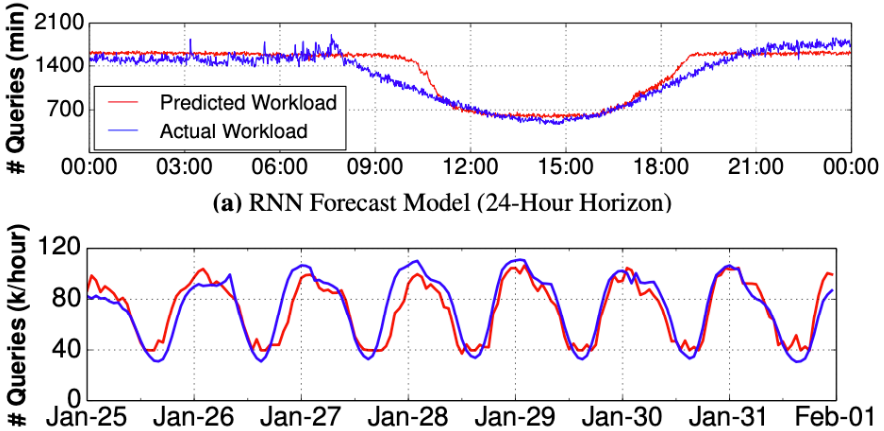


Figure 1: Peloton Self-Driving Architecture – An overview of the forecasting and runtime workflow of Peloton's self-driving components.

Action Plan

- DBMS chooses which one to deploy based on its forecasts, the current database configuration, and objective function. Decision: Cost & Benefit
- RHCM(*receding-horizon control model)*

Peloton-Performance

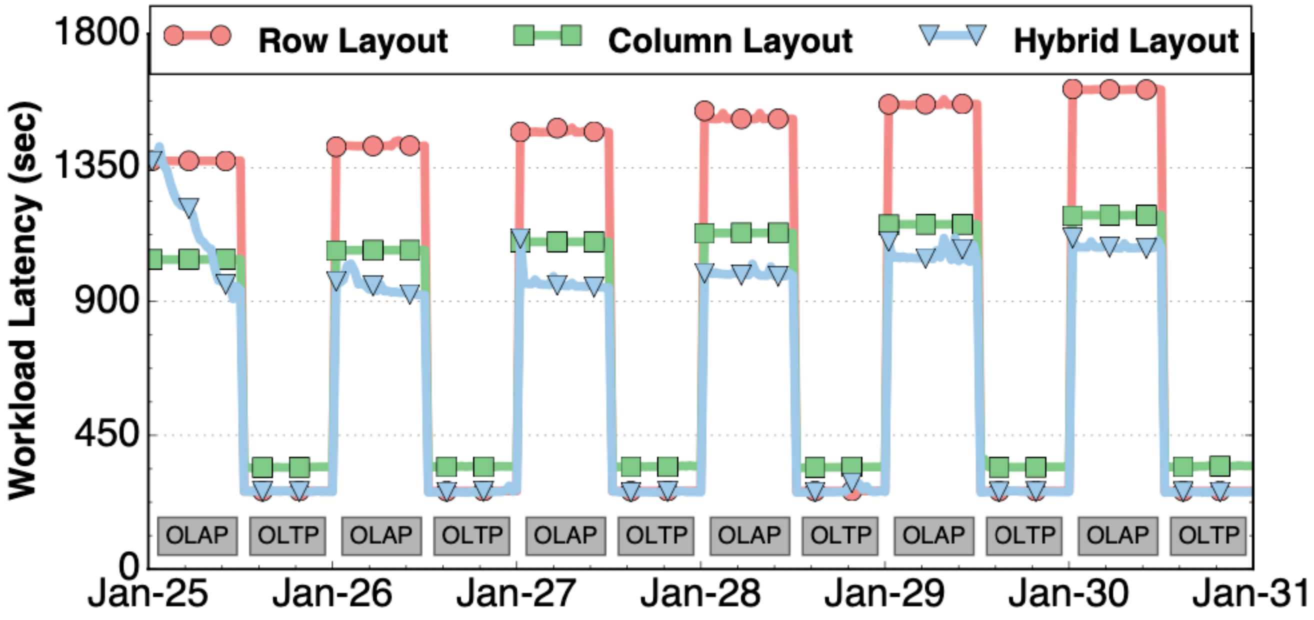


(a) RNN Forecast Model (24-Hour Horizon)

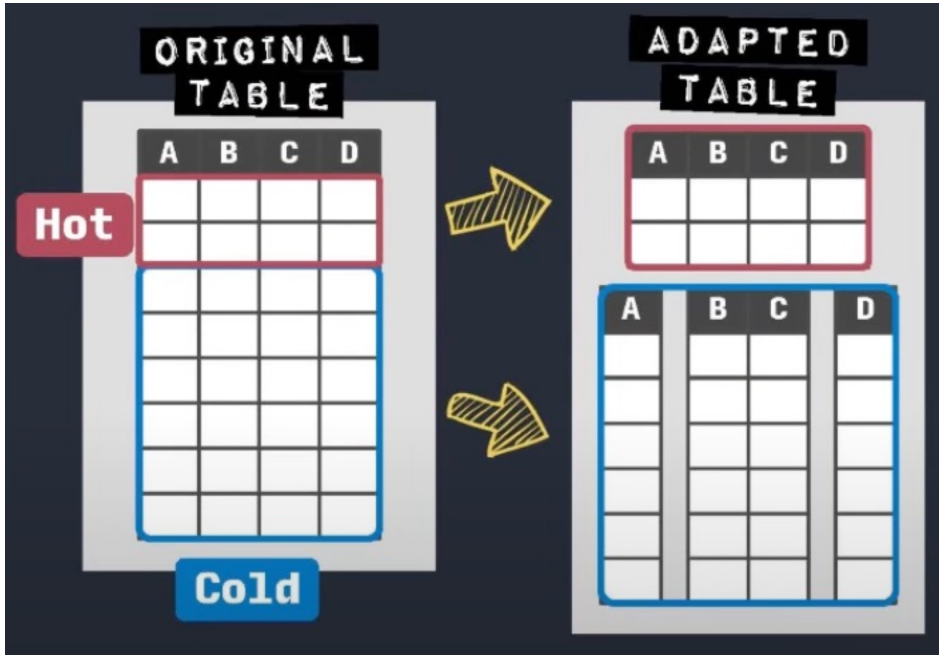
(b) RNN Forecast Model (7-Day Horizon)

	Model (a)	Model (b)
Error Rate	11.3%	13.2%
Training Time	11min	18min
Model Size	2 MB	2 MB
Probe Time	2ms	2ms
Update Time	5ms	5ms

Peloton-Performance



(c) Adaptive Storage Layouts



Questions?