Lecture #11

Special Topics:

Self-Driving Database Management Systems

Workload Modeling I

@Yingjie_Ling // 15-799 // Spring 2022

Previously

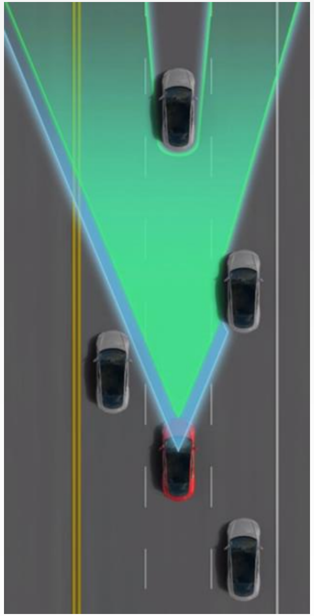
- Index recommendation
- Knob/Parameter Tuning
- Partitioning
- Workload Modeling (Forecasting)

TODAY'S AGENDA

- Motivation
- QB5000
  - Overview
  - Pre-Processor
  - Clusterer
  - Forecaster
- Experimental Analysis
- Parting Thoughts

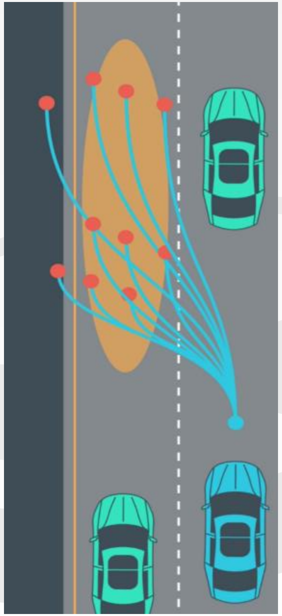
Self-Driving Car

Perception



Action Model

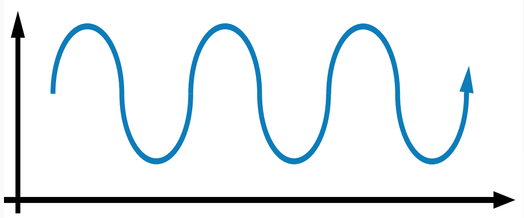
Planning



Source: Lin Ma

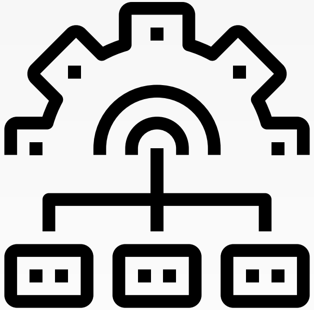
Self-Driving Database

Perception



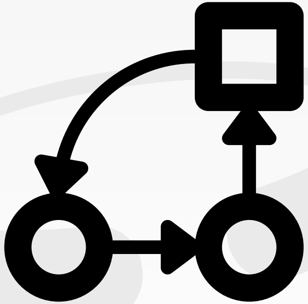
Workload Forecasting

Action Model



Behavior Modeling

Planning



Action Planning

Source: Lin Ma

Workload Patterns

- Bus Tracker: Cycles
- Admissions: Growth and Spikes
- MOOC: Workload Evolution

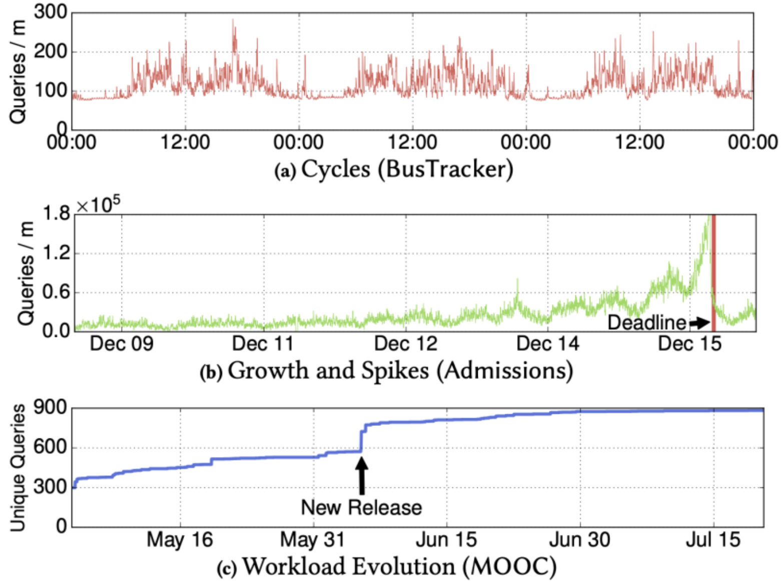


Figure 1: Workload Patterns – Examples of three common workload patterns in database applications.

QUERY-BASED WORKLOAD FORECASTING FOR SELF-DRIVING DATABASE MANAGEMENT SYSTEMS
SIGMOD 2018

TODAY'S AGENDA

- Motivation
- QB5000
- Experimental Analysis
- Parting Thoughts

QB5000 Workflow

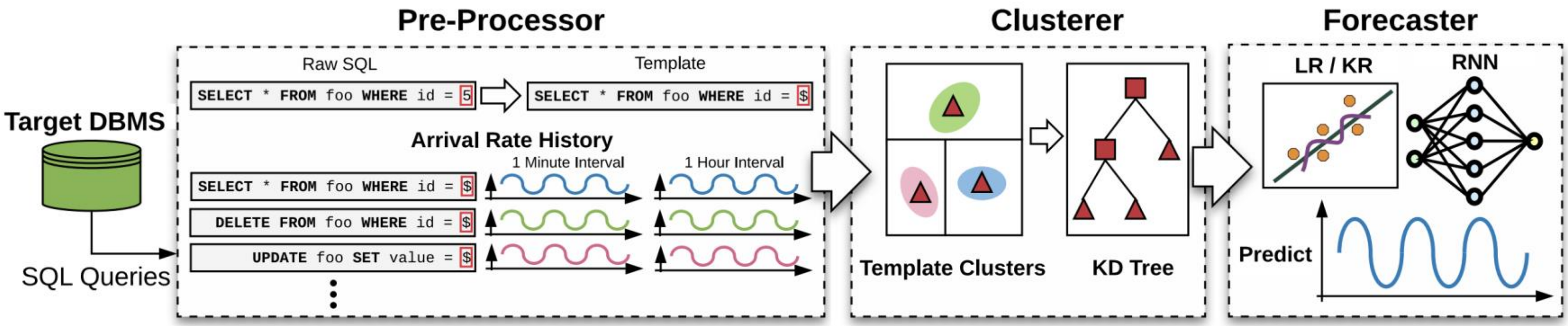


Figure 2: QB5000 Workflow – The framework receives SQL queries from the DBMS. This data is first passed into the Pre-Processor that identifies distinct templates in the workload and records their arrival rate history. Next, the Clusterer combines the templates with similar arrival rate patterns together. This information is then fed into the Forecaster where it builds models that predict the arrival rate of templates in each cluster.

QUERY-BASED WORKLOAD FORECASTING FOR SELF-DRIVING DATABASE MANAGEMENT SYSTEMS
SIGMOD 2018

Pre-Processor

	Admissions	BusTracker	MOOC
Total Number of Queries	2546M	1223M	95M
Total Num of Templates	4060	334	885
Num of Clusters	1950	107	391
Reduction Ratio	**1.3M**	**10.5M**	**0.24M**

Table 2: Workload Reduction – Breakdown of the total number of queries that QB5000 must monitor after applying the reduction techniques in the Pre-Processor and Clusterer.

QUERY-BASED WORKLOAD FORECASTING FOR SELF-DRIVING DATABASE MANAGEMENT SYSTEMS
SIGMOD 2018

Clusterer

- Motivation: still too many templates, not feasible to build a model for each template
- Clusterer must generate stable mapping independent of the state of the database
- Need to be adaptable to the change of workload (when clustered based on time-dependent information)

Clustering Criteria

- Possible similarity features:
  - Physical features

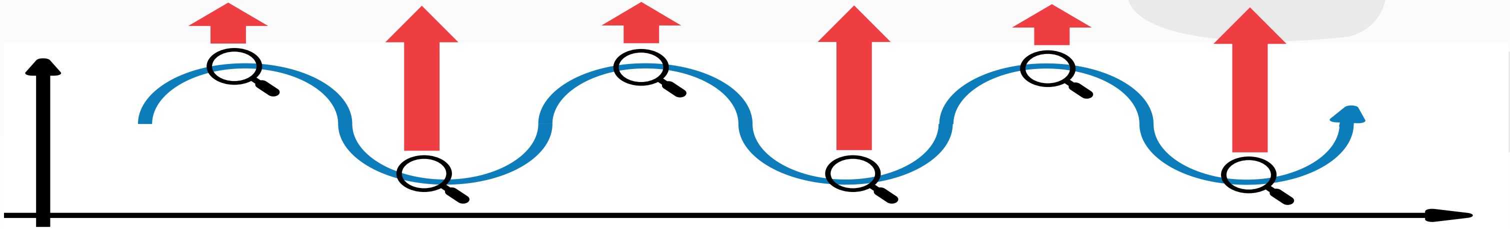
# Tuples Read	# Tuples Write	Latency	…

  - Logical features

Query Type	Columns referenced	# JOINs	…

  - Arrival rate features

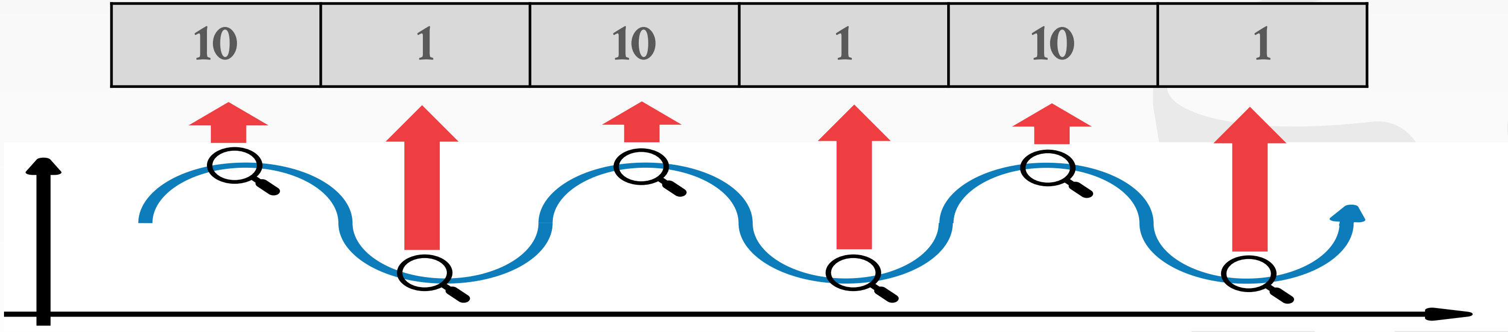
10	1	10	1	10	1



Source: Lin Ma

Clusterer Implementation

- Modified DBSCAN using cluster centers
- Use threshold to determine similarity



Source: Lin Ma

Clusterer Implementation

	Admissions	BusTracker	MOOC
Total Number of Queries	2546M	1223M	95M
Total Num of Templates	4060	334	885
Num of Clusters	1950	107	391
Reduction Ratio	**1.3M**	**10.5M**	**0.24M**

Table 2: Workload Reduction – Breakdown of the total number of queries that QB5000 must monitor after applying the reduction techniques in the Pre-Processor and Clusterer.

QUERY-BASED WORKLOAD FORECASTING FOR SELF-DRIVING DATABASE MANAGEMENT SYSTEMS
SIGMOD 2018

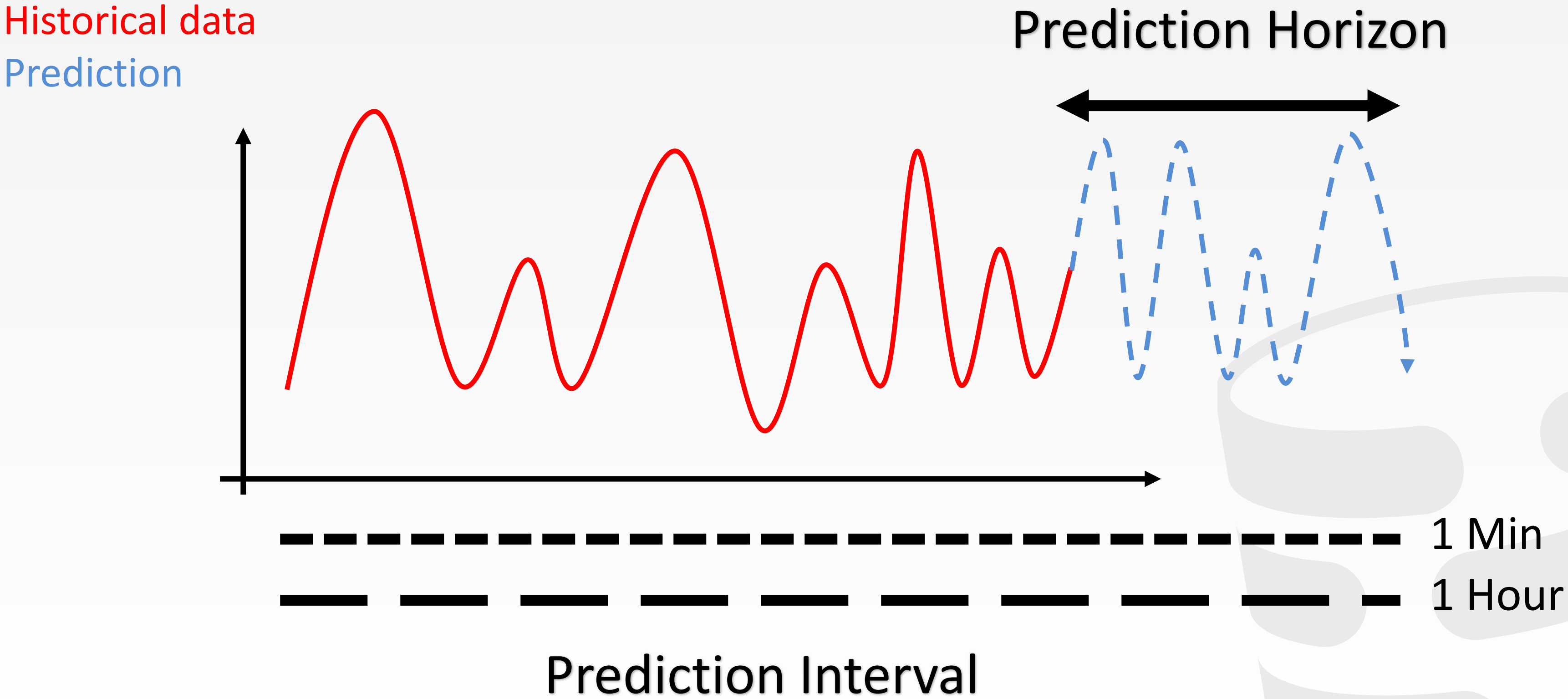
Forecaster

- Each of the top clusters has its own forecasting model
- Combine LR and RNN to build an ensemble for patterns with short periods (ENSEMBLE)
- Use KR for rare but periodic spikes
- Altogether is the HYBRID model
- Trust the KR if its predicted workload volume is above that of ENSEMBLE by more than a threshold

TODAY'S AGENDA

- Motivation
- QB5000
- Experimental Analysis
- Parting Thoughts

Terminology



Source: Lin Ma

Analysis: Number of Clusters

- Efficiency: a small number of clusters capture most of the workload information

- Stability: cluster shift is minimal

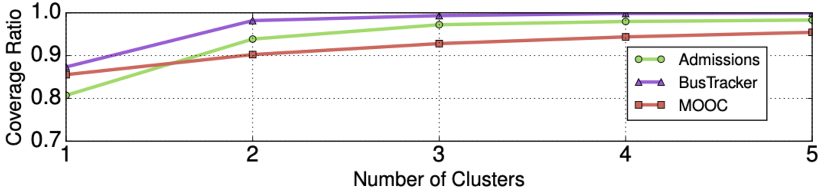


Figure 5: Cluster Coverage – The average ratio between the volume of the largest clusters and the total workload volume.

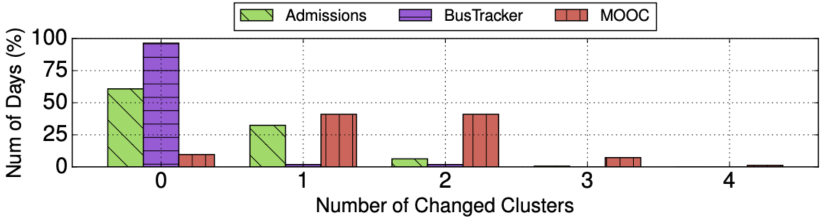


Figure 6: Cluster Change – The number of clusters that changed among the five largest clusters between two consecutive days.

QUERY-BASED WORKLOAD FORECASTING FOR SELF-DRIVING DATABASE MANAGEMENT SYSTEMS
SIGMOD 2018

Analysis: Prediction Accuracy

- LR models perform well for shorter horizons
- RNN models is better for longer horizons
- ENSEMBLE cannot predict rare spikes
- HYBRID is the overall winner by correcting spike predictions with KR

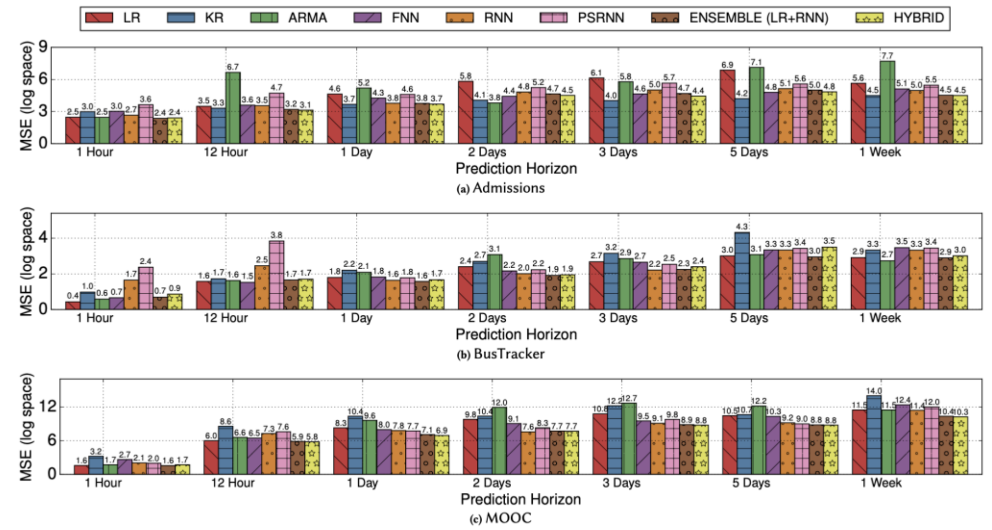
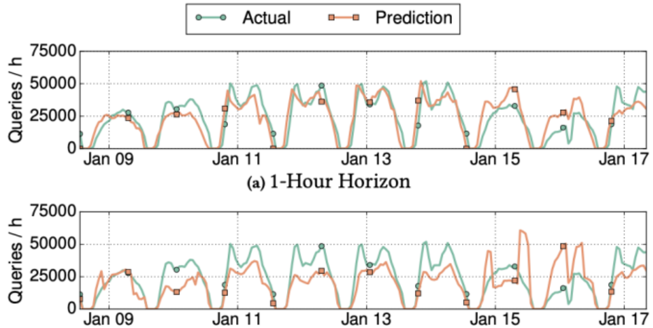


Figure 7: Forecasting Model Evaluation – The average prediction accuracy of the different forecasting models over prediction horizons ranging from one hour to one week for the Admissions, BusTracker, and MOOC workloads.

QUERY-BASED WORKLOAD FORECASTING FOR SELF-DRIVING DATABASE MANAGEMENT SYSTEMS
SIGMOD 2018

Analysis: Prediction Accuracy

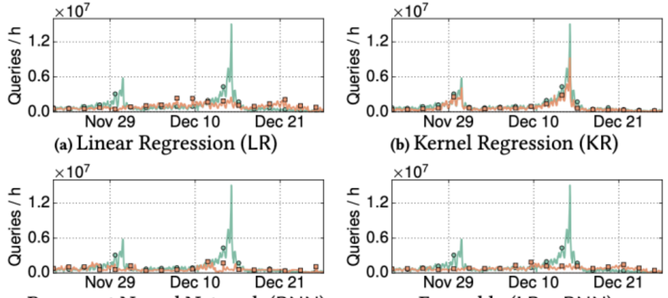
- Prediction for a shorter horizon is more accurate
- Predicted arrival rate pattern matches the actual pattern even for a longer horizon
- Only KR can predict the spike pattern with reasonable accuracy



(a) 1-Hour Horizon

(b) 1-Week Horizon

Figure 8: Prediction Results – Actual vs. predicted query arrival rates for the highest-volume cluster in the BusTracker workload with prediction horizons of one hour and one week.



(a) Linear Regression (LR)

(b) Kernel Regression (KR)

(c) Recurrent Neural Network (RNN)

(d) Ensemble (LR + RNN)

Figure 9: Prediction Results – Actual vs. predicted query arrival rates for the combined clusters in the Admissions workload with spike patterns.

QUERY-BASED WORKLOAD FORECASTING FOR SELF-DRIVING DATABASE MANAGEMENT SYSTEMS
SIGMOD 2018

Analysis: Prediction Interval

- Prediction accuracy is higher for models with shorter interval
- More time consuming to train models with shorter interval
- Models trained with shorter interval are larger and more complex

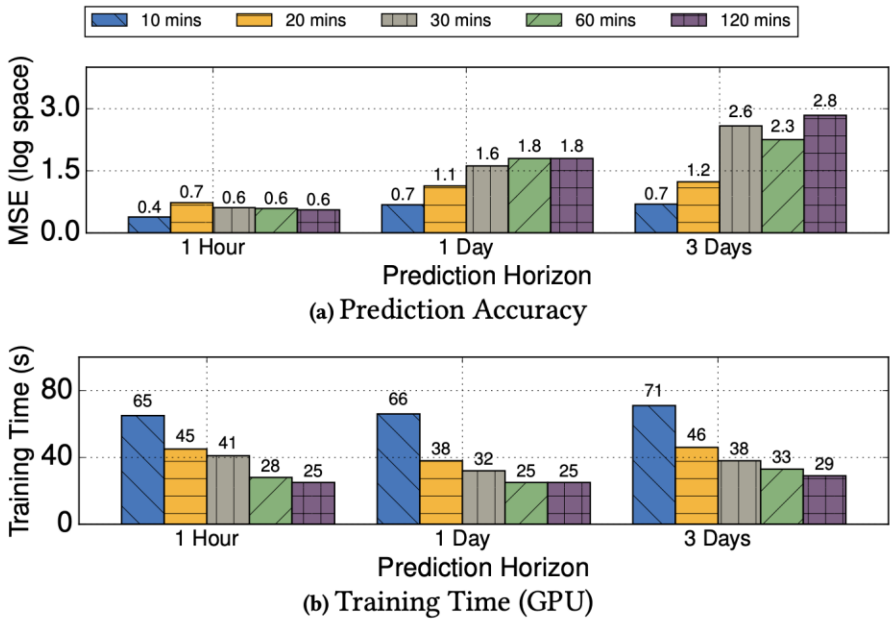


Figure 10: Prediction Interval Evaluation – The average prediction accuracy and training time with different intervals for BusTracker.

QUERY-BASED WORKLOAD FORECASTING FOR SELF-DRIVING DATABASE MANAGEMENT SYSTEMS
SIGMOD 2018

Analysis: Computation & Storage Overhead

- Reasonable storage overhead overall
- Training the RNN models is the most time-consuming
- GPU speeds up training for RNN models
- KR models consumes higher storage

		Pre-Processor	Clusterer	LR	RNN	KR
Computation	Admissions	0.043ms/query	15s/day	GPU:0.3s CPU:0.3s	GPU:9s CPU:58s	GPU:0.16 CPU:0.18s
	BusTracker	0.05ms/query	3s/day	CPU:0.12s GPU:0.13s	GPU:33s CPU:221s	GPU:0.02s CPU:0.02s
	MOOC	0.048ms/query	12s/day	GPU:0.54s CPU:0.51s	GPU:5s CPU:18s	GPU:0.04s CPU:0.04s
Storage	Admissions	1.6MB/day	6.7KB	100B	28KB	11MB
	BusTracker	0.25MB/day	2.2KB	100B	28KB	1.9MB
	MOOC	1.4MB/day	0.8KB	100B	28KB	0.4MB

Table 4: Computation & Storage Overhead – The measurements for QB5000's different components.

Analysis: Automatic Index Selection

- AUTO: candidate indexes generated using the predicted workload of the three largest clusters

- STATIC: same index selection algorithm but applied to a fixed workload sample

- AUTO-LOGICAL: clustered based on logical feature

5x
+28%
-20%

-78%
-23%
~0%

180x
~0%
-20%

-99%
~0%
+38%

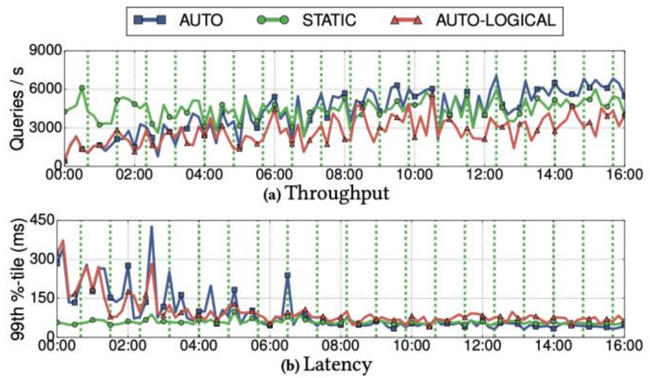


Figure 11: Index Selection (MySQL) – Performance measurements for the Admissions workload using different index selection techniques.

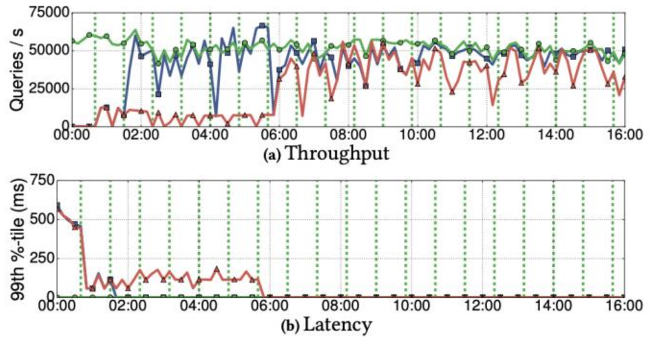


Figure 12: Index Selection (PostgreSQL) – Performance measurements for the BusTracker workload using different index selection techniques.

TODAY'S AGENDA

- Motivation
- QB5000
- Experimental Analysis
- Parting Thoughts

PARTING THOUGHTS

- A better clustering criterion that not only captures the arrival rate but also the semantics?
- Can a single forecaster predict the arrival rate for all/multiple clusters?
- How does QB5000 perform with more challenging workload, such as the MOOC workload?
- How else can the forecaster be used?

References

[1] Lin Ma, Dana Van Aken, Ahmed Hefny, Gustavo Mezerhane, Andrew Pavlo, and Geoffrey J. Gordon. 2018. Query-based Workload Forecasting for SelfDriving Database Management Systems. In Proceedings of 2018 International Conference on Management of Data (SIGMOD'18). ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3183713.3196908.