Point of view	2017[1]	2021[2]	Connection&&relationship
abstract	Main Focus：Latency Peloton includes an embedded monitor that follows the system's internal flow of events for executed queries. Each query entry is annotated with its resource utilization. The DBMS then builds predictive models for the expected workload of the application. It uses these models to identify bottlenecks and other problems (for example, missing indexes, overloaded nodes), and then chooses the best action. The system performs this action while still processing the routine work of the application, loading and collecting new monitoring data to understand how these operations affect its performance.	Main Focus：Latency In 2021, The paper mainly focus on action selection. (Given the predictive workload and potential optimization behaviours, how the system chooses the final optimization method(combination).	In 2017, the CMU team released peloton, which has a fully automated optimization component, dedicated to predicting future workloads and providing optimization recommendations. In the 2017 paper, the CMU team focused on workload classification and prediction. In the 2021 paper, the CMU team focused on how the system can automatically and quickly select the best operation decision after learning the workload prediction and optional optimization methods. The inputs for MB2(2021 paper) are the forecasted workload and a potential action

S1.1 unsupervised learning(cluster classfication)	Purpose: Use unsupervised learning methods to divide queries into different clusters. The purpose is to reduce the number of DBMS prediction models, so that it is easier (and more accurate) to predict application's behavior. Principle: The initial implementation of Peloton uses the DBSCAN algorithm Determinants of cluster grouping 1.runtime 2.semantic The DBMS uses cross validation to determine whether a rebuild is required(the clusters are not realiable any more)	YouTube[3]：Ma Li(PhD in CMU, Main contributor of the paper).The presentation states that the runtime and semantics are not good enough for classification (starting from 16.18 seconds). Lin Ma's opinions is that the furture work should focus on the arrival rate.	In the 2021 paper, the CMU team focused on the MD2 framework, which is dedicated to generating behavior models to predict performance, so that the DBMS can predict future workloads and understand the effects of various optimization operations on its own system. The input of MD2 is workload prediction and optional optimization methods(potential limited,could it invoke an functional API that it haven't seen before????). Otherwise it would not be that automatic if the MD2 only support some common optimization methods.
S1.2 supervised learning	**Purpose**: To train a predictive model that can predict the arrival rate of queries in each cluster. (Ie: the number of queries per cluster at a specific point in time)	Article: No details YouTube: No details	This technical point did not get a detailed description in the 2021 paper.

	Principle: After each query executed by the DBMS, each query is marked by its corresponding cluster identifier, and then a histogram is used to track the number of queries reached by each cluster in a period of time. Peloton uses this data to train predictive models that predict how many queries in each cluster will be executed by the application in the future. Method: RNN, Peloton will use and maintain multiple RNNs, which predict the workload in different time interval. (Q: who to decide the length of time interval)		
S2 optimization	**Purpose**: After the system completes cluster classification and load prediction, **the system will start looking for actions (indexing, hashing) that can improve operational performance**. This search is guided by	**Purpose**: After the system completes cluster classification and load prediction, the system will start looking for actions (add index, hash) that can improve operational performance	In the 2017 paper, the CMU team specifically noted that all optimizable operations (actions) will be stored in the catalog, although in the 2021 paper, CMU simply summarizes these optimizable operations (Action)

	the **predictive model (previous step),** so the system will look for the action that will bring the most benefit. Peloton will store these optimized actions (actions) in a catolog. At the same time, the catolog will also store changes in the system when the operations are called. When the optimization operation is put into the catalog, the DBMS will choose which one to deploy based on the prediction. There is a special method called the **Reverse Horizon Control Model (RHCM)** that can solve this prediction problem. This model is used to manage complex systems such as self-driving cars. The basic idea of RHCM is that in each time period, the system uses forecasts to estimate the workload within a certain limited range. Then search for a	Principle: To avoid the impact of high-dimensional features (high-dimensionality can easily cause over-fitting of the model, that is, the model can only be used for specific scenarios to achieve the greatest benefit). MD2 decomposes the DBMS into multiple independent operating units (OUs). Each OU will help the DBMS to complete a specific task, such as building a join hash table (JHT). Based on different OUs that handle different DBMS detailed tasks, the input feature of each OU may be different, but MD2 can ensure that the output of each OU contains the four most important feature points (CPU, I/O, memory, Time). When all OUs have completed their tasks, the system will use statistical analysis to summarize the prediction results of all OUs, and use the interference model to summarize the OU	as the candidate actions, but due to the relationship and publication order of the two articles. We have reason to believe that in MD2, optimizable actions will be obtained in the catalog. In addition, the biggest difference between 2017 and 21 is to determine the impact of optimization actions on itself. The RHCM is committed to minimizing the objective function within a specific time frame, but this will cause a problem. How far into the future does the system need to be considered? When selecting an action. If you only consider the short timeline in the future, this will make the DBMS too late to prepare for the upcoming load, but using a too long timeline will make its model very slow and unable to deal with unexpected problems.

	series of actions to minimize the objective function (delay).	model predictions.	The OU has solved the impact of high dimensions in a loosely coupled manner. At the same time, the use of statistical analysis and interference models also help the system effectively solve the impact of high concurrency。
S3 deployment	Insufficient information in the paper	Insufficient information in the paper	In the 2017 and 2021 papers, neither of them directly detailed action planning actions. As Lin Ma said on YouTube's presentation, the current CMU team is actively dealing with this issue. (40.47 seconds) In YouTube, Lin Ma proposes a pilot mechanism. When the predictive workload and behavior model are obtained, the pilot will be automatically selected and deployed.

Reference

[1]: https://db.cs.cmu.edu/papers/2017/p42-pavlo-cidr17.pdf

[2] : https://db.cs.cmu.edu/papers/2021/ma-sigmod2021.pdf

[3] : https://www.youtube.com/watch?v=YqW9Pq5488s