noise page Dotobose Gyms

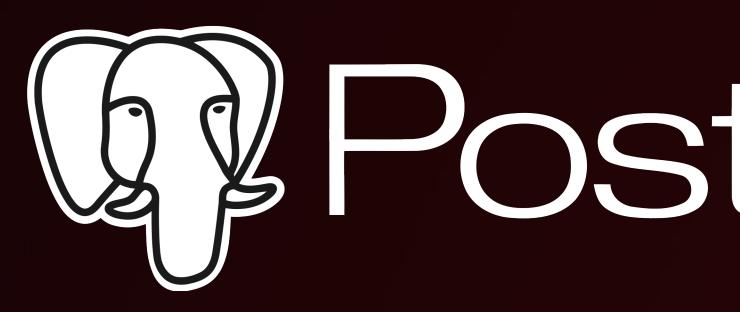
Wan Shen Lim, Matthew Butrovich, William Zhang Andrew Crotty⁺, Lin Ma⁻, Peijing Xu, Johannes Gehrke*, Andrew Pavlo

Carnegie Mellon University Northwestern University **University of Michigan * Microsoft Research**



INTRODUCTION > Overview

Database Gyms: an integrated environment providing a unified API of pluggable components to obtain highquality training data for autonomous DBMSs



PostgreSQL



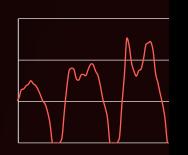
HOLD UP What happened to Peloton and NoisePage?



INTRODUCTION > Self-Driving Databases @ CMU

Ling Zhang

2014 -



Workloa [SIGmod 10]

[JIGINOD ZI]

Zoom Meeting



(monte cano n'ee Search)









SELF-DRIVING DATABASES > Architecture

What components are needed for self-driving?

- Workload forecasting
- Behavior modeling
- Action planning

How have recent papers focused on these problems?

Research Data Management Track Paper	SIGMOD '21, June 20-25, 2021, Virtual Event, China
MB2: Decomposed Be	ehavior Modeling for
Self-Driving Database	
	Wuwen Wang, Matthew Butrovich
Wan Shen Lim, Prashant Carnegie Mell	
{lin.ma,mbutrovi,wanshenl,pmenon,pavlo}@cs.c	
ABSTRACT	embedded low-level models for self-driving operations, nor do they
balakae management systems (DMMS) are notexicasly filled in the high and administ." The goal of a self-articular (DMS) is to remove have impediated in the goal of a self-articular (DMS) is to remove the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the DMS is near and epitian all appeted of the system of the self-article of the self-article of the self-article of the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the self-article of the the self-article of the self-article of the se	embedde low-lower models for well-driving operations, nor ob they appropt personical the training data assolution would will be also the training the second to build and a models. Technologies for constructing database behaviour models (full under dot. Analytical models are a human-devined formula to describe a DBMS component's behaviour, such as the buffer pool or look manager [42, 63, 71]. There models are arrower DBMS and require a structure of the training data assolution of the training of the second structure of the training data assolution of the probability of the second structure of the training of the training area associated associated association of the training of the training of carriers in known (i.e. 64, 72), but and fully the second structure of the part of neutrino of the training data association of the training of the second structure of the training data association (b). The second structure of the second structure of the second of carriers in known (i.e. 64, 72), but and fully the second structure of the data of carriers in known (i.e. 64, 72), but as at data of the second structure of the data of carriers in known (i.e. 64, 72), but as at data of the second structure of the data of carriers in known (i.e. 64, 72), but as at data of the second structure of the data of the second structure of the second structure of the second concernative second as a training of the second structure of the second concernative second as a training of the solution of the second structure o
A welf-shring DNMS can configure, tune, and optimize itself with- our human intervention, can at the application workload, dataset, and operating environment evolve [96]. Such automation asket to environ the configuration and costs involved with DNMS deploy- baction mutants and costs involved with DNMS deploy- baction mutants and costs involved and DNMS deploy- lexiton mutants of the production of the statement of a statement of the statement of the statement of statement of the statement of t	terms into mult, independent operating and (Obb) (e.g., building a hash hash, finding, finding per secords), MBC has neves ML methods, the most second performance of the second second second second resource consumption for the current DBMS state. Compared to a single monsibility model for the current DBMS, these CU-models are performance for the future second second second second second have smaller append dimensions, require less training time, and pro- formance for the future workshop with the second seco

Query-ba	sed Work	oad Foreca	asting for				
Self-Driving Database Management Systems							
Lin Ma	Dana V	an Aken	Ahmed Hefny				
Carnegie Mellon University	Carnegie Mellon University		Carnegie Mellon University				
lin.ma@cs.cmu.edu	dvanaken@	cs.cmu.edu	ahefny@cs.cmu.edu				
Gustavo Mezerhane	Andrey	v Pavlo	Geoffrey J. Gordon				
Carnegie Mellon University	Carnegie Mel	lon University	Carnegie Mellon University				
gangulo@andrew.cmu.edu		s.cmu.edu	ggordon@cs.cmu.edu				
ABSTRACT		activities [15]. But	personnel is estimated to be 50% of a DBMS				
The first step towards an autonomous database m	anagement system	total cost [45]. If the	DBMS could optimize itself automatically, then				
(DBMS) is the ability to model the target appli			my of the complications and costs involved with				
This is necessary to allow the system to antic			40]. Such a "self-driving" DBMS identifies which				
load needs and select the proper optimizations i			ould optimize without human intervention.				
Previous forecasting techniques model the reso	urce utilization of		asons why new efforts to develop a self-driving				
the queries. Such metrics, however, change whe			ng even though other attempts have been less				
design of the database and the hardware resource	es change, thereby		remost is that the improved capabilities of mod uputational hardware enable the DBMS to collect				
rendering previous forecasting models useless.			iputational hardware enable the DBMS to collect its behavior and then use it to train machine				
We present a robust forecasting framework ca that allows a DRMS to predict the amounted and			its behavior and then use it to train machine is that are more complex than what was possi				
that allows a DBMS to predict the expected arrival rate of queries in the future based on historical data. To better support highly		ble before. The second reason is that the recent advancements in					
dynamic environments, our approach uses the b			ep neural networks and reinforcement learning				
of queries in the workload rather than the amount of physical re- sources used for query execution. It provides multiple horizons		will allow the DBMS to continually improve its models over time as it learns more about an application's workload.					
				(short- vs. long-term) with different aggregation			omous, the DBMS must be able to predict what
present a clustering-based technique for reducing the total num- ber of forecasting models to maintain. To evaluate our approach, we compare our forecasting models against other state-of-the-art models on three real-world database traces. We implemented our models in an external controller for PostgreSQL and MySQL and demonstrate their effectiveness in selecting indexes,		the workload will look like in the future. If a self-driving DBMS only considers the behavior of the application in the past when selecting which optimizations to apply, these optimizations may be sub-optimal for the workload in the near future. It can also cause re source contention if the DBMS tries to apply optimizations after the workload has hifted (e.g., it is rentring a pask load period). Instead					
				ACM Reference Format:		a self-driving DBM	IS should choose its optimizations proactively
				ACM Reference Format: Lin Ma, Dana Van Aken, Ahmed Hefny, Gustavo Meze	thans Andrew Paulo	according to the expected workload patterns in the future. But the	
				and Geoffrey J. Gordon. 2018. Query-based Workload		DBMS's ability to achieve this is highly dependent on its knowledge	
				Driving Database Management Systems. In Proceedings of 2018 International Conference on Management of Data (SIGMOD'18), ACM, New York, NY, USA,		of the queries and patterns in the application's workload. Previous work has studied database workload modeling in dil	
	as studied database workload modeling in dil r example, one way is to model the demands						
15 pages. https://doi.org/10.1145/3183713.3196908			e system, rather than a direct representation of				
1 INTRODUCTION			[14, 44]. Other methods model the performance				
			swering "what-if" questions about changes in				
With the increasing complexity of DBMSs in m applications, it is more difficult now than eve		OLTP workloads [3	37, 38]. They model the workload as a mixture				
applications, it is more difficult now than eve ministrators (DBAs) to tune these systems to ac			f transactions with a fixed ratio. There is also				
formance. Many DBAs spend nearly 25% of th			v the workload will shift over time using hidder				
			8-31] or regressions [19, 36]. Earlier work has				
Permission to make digital or hard copies of all or part of th	is work for personal or		ase workloads using more formal methods with				
classroom use is granted without fee provided that copies are for profit or commercial advantage and that copies bear this no			tion types and arrival rates [48, 49]. hods have deficiencies that make them inade				
on the first page. Copyrights for components of this work ow	rned by others than the		nods nave deficiencies that make them inade mous system. For example, some use a lossy com				
author(s) must be honored. Abstracting with credit is permitte republish, to post on servers or to redistribute to lists, requires (prior specific permission		at only maintains high-level statistics, such as av				
and/or a fee, Request permissions from permissions@ucm.or	8-	erage query latency	and resource utilization [14, 37, 38, 44]. Other				
and/or a fee. Request permissions from permissions@ucm.or SIGMOD'18, June 10–15, 2018, Housten, TX, USA 12, 2018, Convright held by the ouver/author(s). Publication							
regionance to proceeding the error or or originary model on the second s		assume that the too they only generate	and resource utilization [14, 37, 38, 44]. Others l is provided with a static workload [48, 49], oo new models when the workload shifts, thereby how the volume of queries and the workload				

Plan-Structured Deep Neural Network Models for Query Performance Prediction			
Ryan Marcus Brandeis University	Olga Papaemmanouil Brandeis University		
ryan@cs.brandeis.edu	olga@cs.brandeis.edu		
AUSTRACT Query performance prediction, the task of predicting a query's la- tion of the second of the second of the second of the second material of the second of the second of the second of the performance of the second of the second of the second tasks and performance of the second second of the sec	From approaches have focused en hand-doigning neu profile tre memory (e.g., 11, 45); nammelly distring minimum (e.g. 12), and the second second second second second second second regulate level and generate (e.g., 153), for combin- ing plate level and generate (e.g., 153), for combin- ing plate level and generate (e.g., 153), for combi- site (e.g., 153), for the second second second second second regulate effect for neutral second second second second triangle and the second secon		
1. NURROUGLING: Conceptionance prediction (QPT), the task of predicting the dimmagnetic tasks, including admission could [67], resource management [64], and maintaining [54]. All [1, 34], QPT is also no a mathematical strain and the second prediction of the prediction of the second prediction of the second parameter of fatters. An other according the torow, the under- loging data distribution, and the resource parallelisty. As database imagingly styles and the second prediction only gets under difficult free wide direction of the second parameters of the second prediction of the second prediction only gets under difficult free wide direction of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters of the second parameters. The second parameters of the second parameters of the second parame	of query performance performance, provident and the temperature of the second human deviced method method in the second human deviced method method in the second human deviced method in the second human device provides the second human device performance and the second human device performance and the second human device and the second		



DBMind: A Self-Driving Platform in openGauss

and a database instance and netros the collected information in a three power has an even idea or inserves dualbase integration that we want idea or inserves dualbase integration that we even in the serve and its (v) (v) integrating genomeral indication of the power is the server in the server is the serve * These authors contribute equally to this work

A Unified Transferable Model for ML-Enhanced DBMS

Ziniu Wu², Pei Yu^{1,4,#}, Peilun Yang^{1,3,#}, Rong Zhu^{1,*}, Yuxing Han¹, Yaliang Li¹, Defu Lian⁴, Kai Zeng¹, Jingren Zhou¹ aba Group, ²Massachusetts Institute of Technology, chnology Sydney, ⁴University of Science and Tehnology, China w@mit.edu, {yangpeilun.ypl, yupei.yu, red.zr, yaliang.li, zengkai.zk

ABSTRACT Recently, the database management system (DBMS) com-	of data volume and complexity, it becomes increasingly dif- ficult to maintain DBMS purely using human efforts.
muity has witnessed the power of michine learning (ML) solutions for DBNs task. Donget the they promising perfor- sion of the second start of the second start of the second infectory. First, these ML-based methods in DBMS are not effective enough because they are optimized on each specific task, and cannot explore or understand the initiatic connec- tion of the second start of the second start of the second constraints of the second start of	Recently, the prosperity of machine learning (ML), es- product deep learning, highly to revolve a large mather of product deep learning. In plot 1 and
ing power and viability of MTALLF, we provide a concrete and very promising case study on query optimization tasks. Last but not least, we discuss several concrete research op- portunities along this line of work.	be not explore the nonvening tradification means to an near to impractical solutions and/or ineffective models. Transferability across databases: Existing ML methods for DBMS only focus on learning the database-specific knowledge and ignore the database-agnostic meta knowledge that can be transferred to new DBs. Therefore, the need to
1. INTRODUCTION	that can be transferred to new DBs. Therefore, they need to retrain the entire model from scratch for a new DB, and gen-
Database management system (DBMS) is the cornerstone of a broad range of applications such as big data platforms, cloud computing, internet of things, and artificial intelli-	erally require an excessive and impractical amount of data, such as executed queries and logs, for each retraining, which is very expensive to acquire especially for a new DB [Ma et al. 2020] (referred to as the notroinus "cold-start" prob-
gence. Designing and tuning DBMS involves a series of complicated tasks ranging from physical design, configura- tion tuning, to query optimization and execution schedul- ing, which all require intensive expertise. With the growth	lem). Fortunately, some meta knowledge can be distilled and shared across DBs to mitigate this problem. This knowledge (such as expert experience and heuristics in the physical join implementation and access path selection) is independent of each specific DB. For example, the query optimizer usually chooses an index scan for high-selectivity predicates and a
# These two authors contribute equally to this paper. * Corresponding author.	sequential scan for low-selectivity ones; and the hash join is usually more memory-intensive than nest loop join and
4 Corresponding attains: This article is published under a Creative Commons Attribution License thttp://creativecommons.org/licenses/by/3/0/, which permits distribution vision and an analysis of the second second second second second vision and compression on however bar authority and CIDB 2022. 12th Annual Conference on Instructure Data Systems Research (CIDR '22) Jannery 9-12, 2022. Chaminade, USA.	is usually more memory-intensive than nest toop join and merge join. This knowledge should be distilled and shared across various databases to avoid the redundant learning process and mitigate the 'cold-start' problem. (2) Transferability across tasks: Existing ML approaches are only optimized on individual DBMS tasks and neglect

An End-to-End Automatic Cloud Database Tuning System Using Deep Reinforcement Learning

ABSTRACT	our model and improves efficiency of online tuning. We con		
Configuration tuning is with to optimize the performance of database magnetic stytem (DMSN). It becomes more tables and urgent for cloud databases (CDD) due to the di- yerse database interactions and performance in the di- transmission of the distribution of the distribution of the performance in an odd-over an unner-trait, they adopt to performance in an odd-over an unner-trait, they adopt to provide the distribution of the distribution of the performance in an odd-over an unner-trait, they adopt to the distribution of the distribution of the distribution of the performance in an odd-over an unner-trait, the distribution of the distribution of the distribution of the distribution of the performance in a start of the distribution of the	our model and improves efficiency of online tuning. We con- ducted estematics experiments under 6 different workloads (CBFure, Experimental results showed that CBFure had, good adaptability and significantly outperformed the state of the art tuning tools and DBA experts. 1 NETOCION The spectrometal results showed that CBFure had, good adaptability and adjust and the state of the art tuning tools and DBA experts. 2 NETOCION The spectrometal of clashese management system (DBAS fields on hundred) of tunble configurations hads. Support for knobs estings can improve the performance for DBAS (e.g., higher throughput and lower latency). However, only database (CDB), however, even the most experiment DBAS database (CDB), however, even the most experiment DBA database (CDB), however, even the most experiment DBA and the of stating appropriate kabo Semigrations. In a close database even provides are foring as challenge that they have to tune cloud database systems for a large num they of users with limited and experiment DBA experts. A a result, developing effective systems to a compliab hat that parameters configurations and experimentials have that parameters configurations and explorations.		
	based methods [4, 14, 35]. The search-based methods, to BestConfig (55), search the optimal parameters based certain given principles. However, they have two limitatis firstly, they spend a great amount of time on searching optimal configurations. Secondly, they restart the search to utilize knowledge gained from previous tuming effort The learning-based methods, e.g., OtterTume [4], with methine-learning techniques to collect, process and anal		
	knobs and recommend possible settings by learning DBA's experiences from historical data. However, they have four limitations. Firstly, they adopt a pipelined learning model which andfers from a severe problem that the optimal solution of the previous stage cannot guarantee the optimal solution in the latter stage and different stages of the model may not work well with each other. Thus they cannot optimize		

y Performance Prediction for Concurrent Queries using Graph Embedding





SELF-DRIVING DATABASES > Modern ML

ML = Models + Training Data

If you do know what model you want,

• 1 month, 1k LOC

If you do <u>not</u> know what model you want,

- Foundation models
- Automated model design (e.g., AutoML)



SELF-DRIVING DATABASES > Modern ML's Implications

Training data is what matters today

With systems knowledge, generate training data that is

- Better (higher quality)
- Faster (less time taken)

Leave the ML to the ML people



TRAINING DATA How do we collect training data?

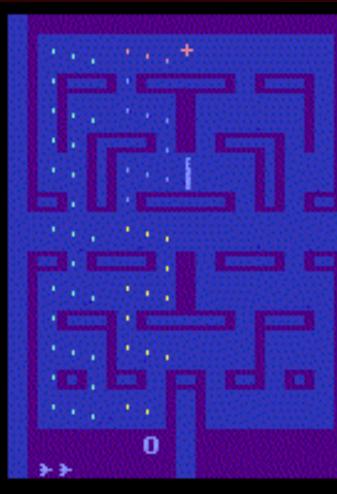


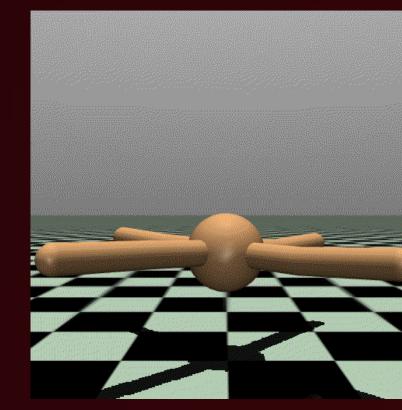
TRAINING DATA > **Obtaining Training Data**

Choice #1: Do it live!

 Performance degradations in production Choice #2: Simulator (via a gym)

- Approximates the behavior of an entity when it would otherwise be too costly, time-consuming, or dangerous to experiment on the real system
- Packaged into toolkits for developing and evaluating different models and algorithms







Ο

0

TRAINING DATA > **Obtaining Training Data via Simulation**

Building a DBMS simulator is difficult

- Key idea: Use the DBMS to simulate itself Requires solving systems and ML problems hand-in-hand We call this integrated solution the Database Gym

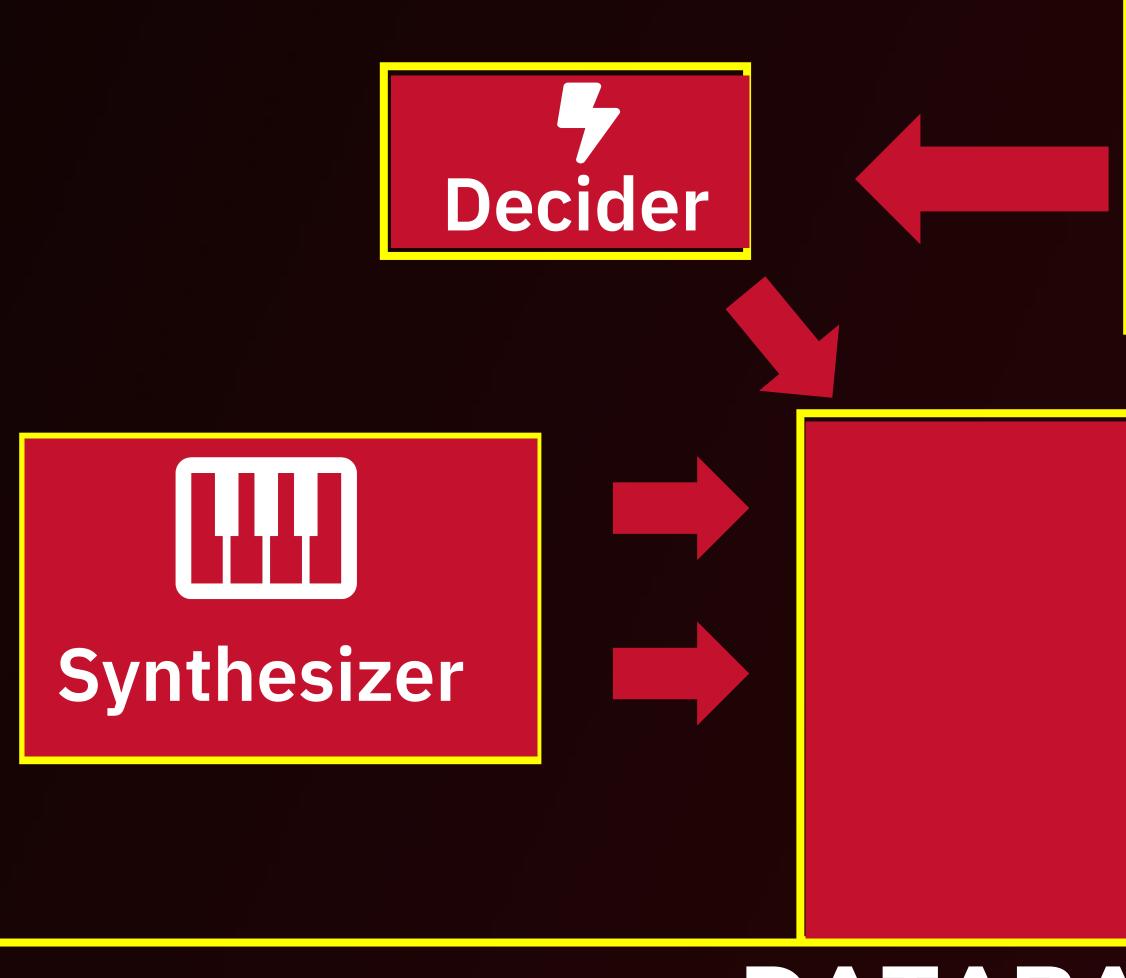


oage

DATABASE GYM Architecture



DATABASE GYM> Using the DBMS as a Simulator



Planner

Image: Constraint of the second sec

DATABASE GYM



DATABASE GYM> Synthesizer

Manages and manipulates the inputs to the database gym Snapshot : backups (e.g., pg_dump)

- Workload : timeseries of SQL queries

Goal: what-if scenarios without replaying the workload • Example: "create a snapshot with 2x the data and increase the queries in the workload by 5x"



DATABASE GYM> Trainer

Given the workload and snapshot, coordinate workload execution to produce training data

- Observability
- Execution

Workload replay tools

pgreplay supports speed factor for replay



DATABASE GYM> Trainer

Choice #1: Foreign Data Wrappers

Save on storage

Choice #2: RAMDISK

Save on disk access time

Choice #3: Query Progress Estimation Save on query execution time

Scan Intercept read()

on-demand

Compute tuples

Scan

200001





DATABASE GYM > Panner

Suggests a list of promising actions

Extensible Rule-based Action Generation

- Inspired by query optimizers (Exodus, Starburst)
- Example rule: "columns that occur together in a WHERE clause"

Snapshot Ο Workload **Training Data**







DATABASE GYM> Decider

Pick the best action out of a list of candidate actions

By building on the OpenAI Gym, this component is free! Leverage what already exists in the ML community Various RL libraries integrate easily with gyms.



CONCLUSION > Takeaways

Stop spending our time on ML problems, focus on database problems

Database Gym: systems for machine learning for systems





Hit Me Up

@ @capybara@discuss.systems

