



Database 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 high-quality training data for autonomous DBMSs



HOLD UP

What happened to Peloton and NoisePage?

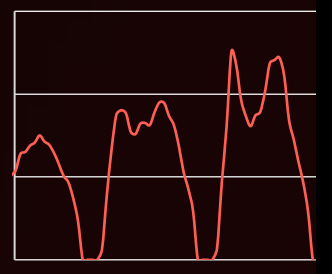
INTRODUCTION >

Self-Driving Databases @ CMU

2014 -



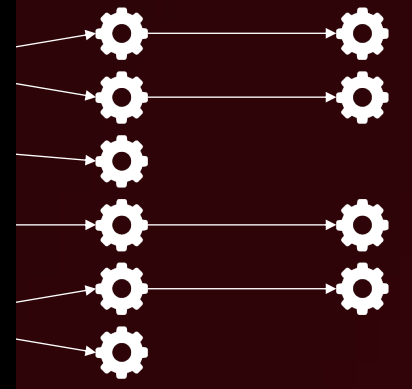
PostgreSQL



Workload
[SIGMOD 10]

[SIGMOD 11]

Planning
(Mobile Carpool Free Search)



Self-Driving Capabilities

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 Task Paper

MBSD2 '21, June 20-21, 2021, Virtual Event, China

MB2: Decomposed Behavior Modeling for Self-Driving Database Management Systems

Liu Ma, William Zhang, Ji Jian, Wuxian Wang, Matthew Butrovich, Wan Shou Lin, Prashant Mishra, Andrew Pavlo

lma@mbd.ac.cn, wzhang@mbd.ac.cn, wjjan@mbd.ac.cn, wwxian@mbd.ac.cn, matthewb@mbd.ac.cn, wanlin@mbd.ac.cn, pmishra@mbd.ac.cn, apavlo@mbd.ac.cn

ABSTRACT

Self-driving database management systems (SDDBMS) are automatically difficult to design and evaluate. This paper presents a self-driving SDDBMS to address these challenges by decomposing behavior modeling into three components: workload forecasting, behavior modeling, and action planning. We present a self-driving SDDBMS architecture that decomposes behavior modeling into three components: workload forecasting, behavior modeling, and action planning. We present a self-driving SDDBMS architecture that decomposes behavior modeling into three components: workload forecasting, behavior modeling, and action planning.

1 INTRODUCTION

A self-driving SDDBMS can manage, tune, and optimize itself without human intervention and can adapt to changing workloads and hardware configurations. These capabilities are essential for a self-driving SDDBMS to meet the requirements of modern data centers. In this paper, we present a self-driving SDDBMS architecture that decomposes behavior modeling into three components: workload forecasting, behavior modeling, and action planning.

Query-based Workload Forecasting for Self-Driving Database Management Systems

Liu Ma, Dana Yan Alan, Ahmed Hefny, Wan Shou Lin, Prashant Mishra, Andrew Pavlo, Gustavo Merchante, Pedro Paulo, Geoffrey J. Gordon

lma@mbd.ac.cn, dyan@mbd.ac.cn, ah@mbd.ac.cn, wanlin@mbd.ac.cn, pmishra@mbd.ac.cn, gggordon@mbd.ac.cn

ABSTRACT

Query-based workload forecasting (QWF) is an essential component of a self-driving database management system (SDDBMS). In this paper, we present a QWF architecture that decomposes workload forecasting into three components: workload forecasting, behavior modeling, and action planning.

1 INTRODUCTION

Query-based workload forecasting (QWF) is an essential component of a self-driving database management system (SDDBMS). In this paper, we present a QWF architecture that decomposes workload forecasting into three components: workload forecasting, behavior modeling, and action planning.

Plan-Structured Deep Neural Network Models for Query Performance Prediction

Ryan Marcus, Olga Papaemmanouli, Ryan Marcus, Olga Papaemmanouli

rm@cs.utoronto.ca, opa@cs.utoronto.ca

ABSTRACT

Plan-structured deep neural network models (PNDNN) are a new class of neural network models for query performance prediction. They are designed to capture the complex dependencies between query plans and their execution performance.

1 INTRODUCTION

Plan-structured deep neural network models (PNDNN) are a new class of neural network models for query performance prediction. They are designed to capture the complex dependencies between query plans and their execution performance.

DBMind: A Self-Driving Tool for Join Graphs in OpenGauss

Xuanbo Zhou^{1*}, Luanyun Yang^{1*}, Jun Sun², Xinyang Zhao¹, Xiang Yu¹, Jiahua Feng¹, Shifu Li^{1*}, Tianjing Wang¹, Xun Li¹, Jingping Liu¹

*Department of Computer Science, Tsinghua University, Beijing, China

zbxuanbo@tsinghua.edu.cn, yangluanyun@tsinghua.edu.cn, sunjun@sem.ac.cn, zhaoxinyang@sem.ac.cn, yuxiang@sem.ac.cn, fengjiahua@sem.ac.cn, lishifu@sem.ac.cn, wangtianjing@sem.ac.cn, lixun@sem.ac.cn, liujingping@sem.ac.cn

ABSTRACT

DBMind is a self-driving tool for join graphs in OpenGauss. It is designed to help database administrators optimize query performance by automatically generating and evaluating join plans.

1 INTRODUCTION

DBMind is a self-driving tool for join graphs in OpenGauss. It is designed to help database administrators optimize query performance by automatically generating and evaluating join plans.

A Unified Transferable Model for ML-Enhanced DBMS

Ziniu Wu¹, Pei Yu^{1*}, Peilin Yang^{1*}, Rang Zhou¹, Xinyang Han¹, Yifang Li¹, Diyu Li^{1*}, Kai Zeng¹, Jingran Zhou¹, Abbas Gholami², Massachusetts Institute of Technology, University of Technology Sydney, University of Science and Technology of China, zniu@mit.edu, yangpeilin@ustc.edu.cn, yangpeilin@ustc.edu.cn, zhouyifang@ustc.edu.cn, hanxinyang@ustc.edu.cn, lifang@ustc.edu.cn, lidiyu@ustc.edu.cn, zengkai@ustc.edu.cn, zhoujingran@ustc.edu.cn, gholami@mit.edu

ABSTRACT

A unified transferable model for ML-enhanced DBMS is presented in this paper. It aims to improve the performance of database management systems by leveraging machine learning techniques across different database environments.

1 INTRODUCTION

A unified transferable model for ML-enhanced DBMS is presented in this paper. It aims to improve the performance of database management systems by leveraging machine learning techniques across different database environments.

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

Ji Zhang¹, Ya Liu¹, Ke Zhu^{2*}, Guohua Li¹, Zhi Yao¹, Bo Cheng¹, Junhui Xing¹, Yongtao Ye¹, Tiancheng Chen¹, Li Liu¹, Miaoqi Kou¹, and Zhenfeng Li¹

¹Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, China

zhangji@whu.edu.cn, liuyaya@whu.edu.cn, zhuke@whu.edu.cn, liguohua@whu.edu.cn, yaozhi@whu.edu.cn, chenbo@whu.edu.cn, xingjunhui@whu.edu.cn, yeyongtao@whu.edu.cn, chentiancheng@whu.edu.cn, liuliu@whu.edu.cn, koumiaoqi@whu.edu.cn, liuzhenfeng@whu.edu.cn

ABSTRACT

An end-to-end automatic cloud database tuning system using deep reinforcement learning (DBTune) is presented. It automates the process of database tuning by learning from historical data and adjusting parameters in real-time.

1 INTRODUCTION

An end-to-end automatic cloud database tuning system using deep reinforcement learning (DBTune) is presented. It automates the process of database tuning by learning from historical data and adjusting parameters in real-time.

Query Performance Prediction for Concurrent Queries using Graph Embedding

Xuanbo Zhou, Ai Sun, Quping Li, Jianhua Feng, Quping Li, Ai Sun, Quping Li, Jianhua Feng

xzhou@mbd.ac.cn, asun@mbd.ac.cn, liquping@mbd.ac.cn, jhfeng@mbd.ac.cn

ABSTRACT

Query performance prediction for concurrent queries using graph embedding (QPGE) is a novel approach to predict the performance of concurrent queries by embedding query plans into a graph structure.

1 INTRODUCTION

Query performance prediction for concurrent queries using graph embedding (QPGE) is a novel approach to predict the performance of concurrent queries by embedding query plans into a graph structure.

page 6

Modern ML

ML = **Models** + **Training Data**

If you do know what model you want,

- 1 month, 1k LOC

If you do not know what model you want,

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

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?

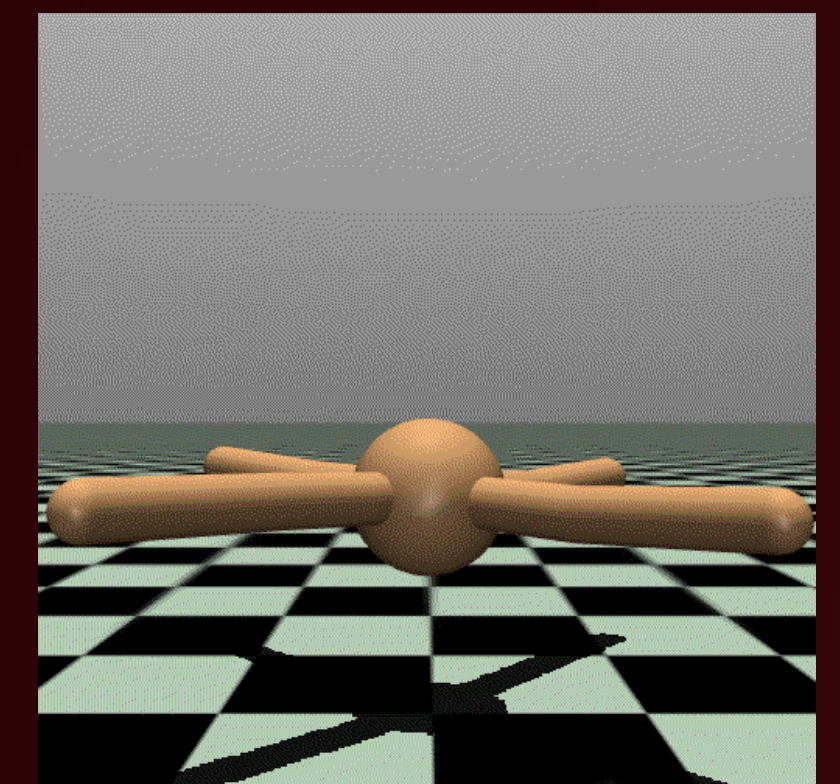
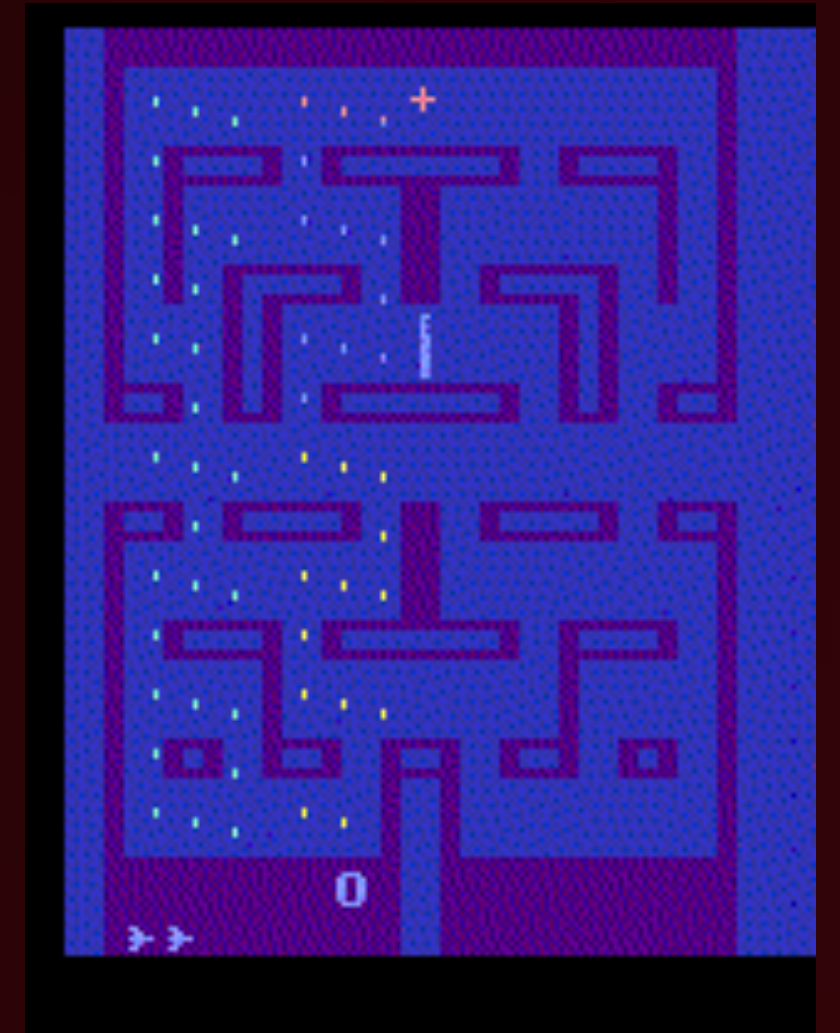
Obtaining Training Data

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



Obtaining Training Data via Simulation

Building a DBMS simulator is difficult

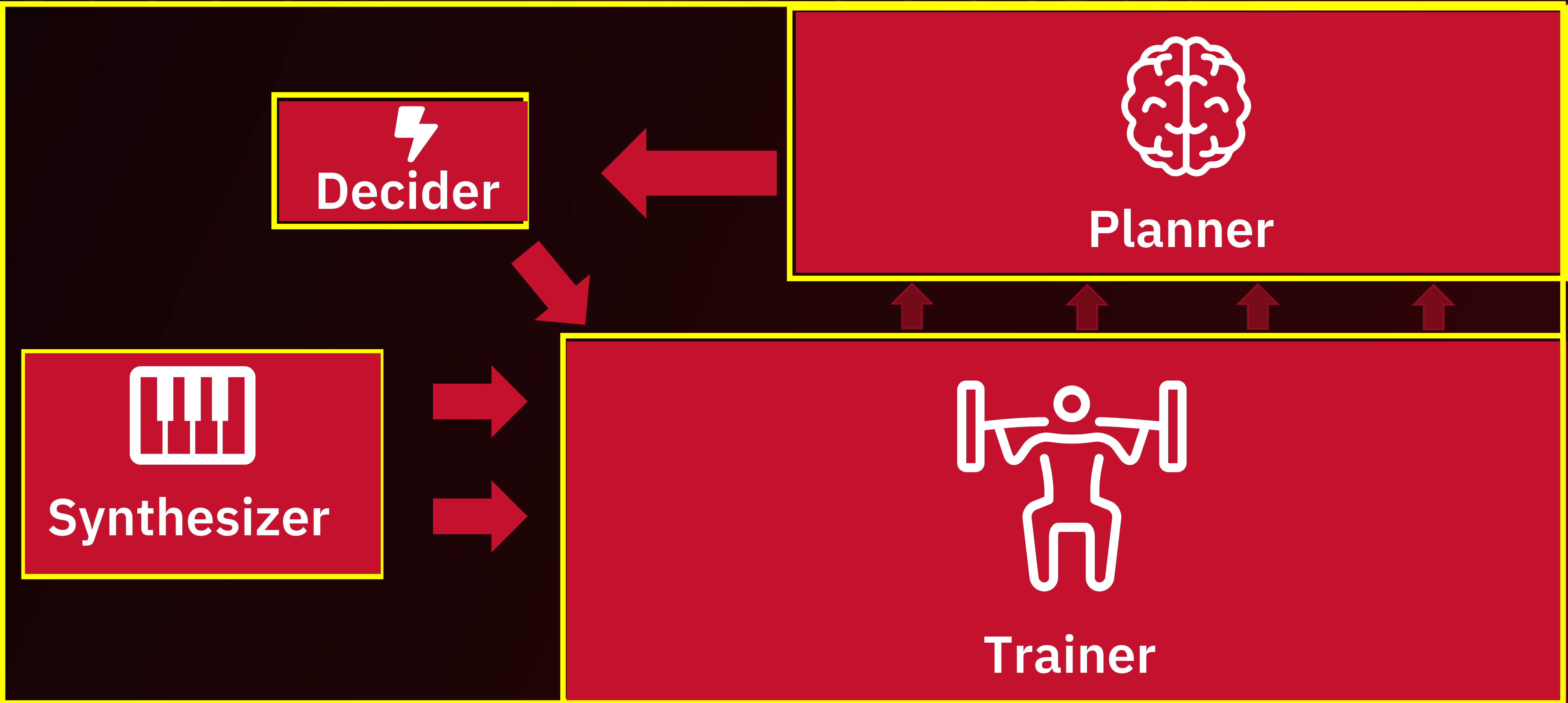
Key idea: Use the DBMS to simulate itself

- Requires solving systems and ML problems hand-in-hand
- We call this integrated solution the Database Gym

DATABASE GYM

Architecture

Using the DBMS as a Simulator



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”

Trainer

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

- Observability
- Execution

Workload replay tools

- pgreplay supports speed factor for replay

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

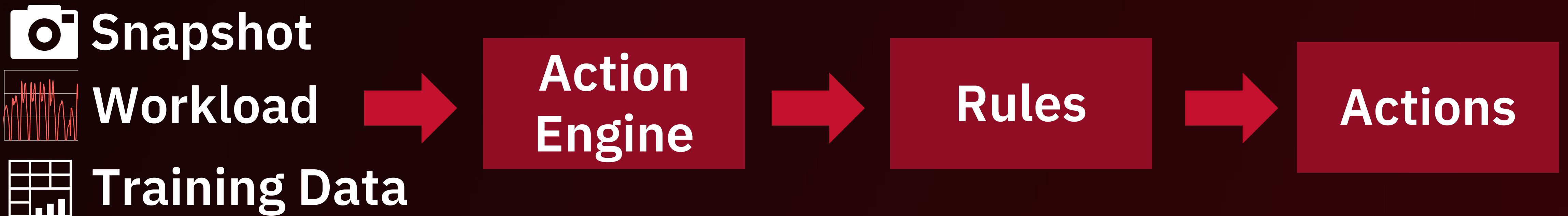


Planner

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”



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

 wanshenl@cs.cmu.edu

 [@capybara@discuss.systems](https://discord.com/users/capybara@discuss.systems)