Self-Driving Databases & My Pregnant Wife: The Hard Parts

@andy_pavlo
I never took a ML course.
Everything I know is from YouTube, research papers, and blogs.

I never took sex-ed.
Everything I know is from a 1993 episode of Perfect Strangers.
I never took sex-education.

Everything I know is from a 1993 episode of Perfect Strangers.

I never took a ML course.

Everything I know is from YouTube, research papers, and blogs.
Autonomous Database Systems

Remove the burden of managing DBMSs from humans.

Target Tasks:
- Physical Database Design
- Knob Configuration Tuning
- Query Plan Tuning
- Capacity Planning / Hardware Provisioning
Autonomous Database Systems

1970s: Self-Adaptive Systems

1990s: Self-Tuning Systems

2010s: Learned Components
Autonomous Database Systems

1970s: Self-Adaptive Systems

1990s: Self-Tuning Systems

2010s: Learned Components
Autonomous Database Systems

1970s: Self-Adaptive Systems

1990s: Self-Tuning Systems

2010s: Learned Components
Autonomous Database Systems

1970s: Self-Adaptive Systems

1990s: Self-Tuning Systems

2010s: Learned Components
Learned Indexes
Learned Indexes

- Key
- Location
- B+Tree
- Neural Net

Learned Scheduling

- Transaction
- Partitioning Scheme
- Location
Learned Indexes

Key \rightarrow B+Tree \rightarrow Location

Key \rightarrow Neural Net \rightarrow Location

Learned Scheduling

Transaction

Unsupervised Clustering
Learned Indexes

Learned Scheduling

Key → B+Tree → Location

Key → Neural Net → Location

Transaction → Unsupervised Clustering

```
SELECT * 
FROM A, B, C 
WHERE A.id = B.id 
AND A.id = C.id
```
Learned
Indexes

B+Tree

Location

Key

Learned
Planning

Histograms

Sketches

Neural Net

Learned
Indexes

Key

Location

B+Tree

Learned
Planning

SELECT *
FROM A, B, C
WHERE A.id = B.id
AND A.id = C.id

1. Introduction

Most modern database systems employ proprietary query optimizers to determine the best plans to execute SQL queries. This paper describes the execution of a SQL query optimizing system that was developed for the VLDB 2001 Conference.
Self-Driving DBMS

A system that configures, manages, and optimizes itself for the target database and its workload without any human input beyond the initialization parameters.

- Objective Function (throughput, latency, cost, energy)
- Constraints (SLOs, resources, cost)

The system is not allowed to make any change that requires a human to make a value judgement.
Self-Driving DBMS Tenets

1. Explore unknown configurations.
2. Generalize over environments / applications / hardware.
3. Reason about delayed consequences.
The Hard Parts

State Modeling

Training Data Collection

Reward Observation
1. STATE MODELING
Succinctly & accurately represent the environment state.
#1 – KB

#2 – Databases
Stochastic
Cannot easily look inside to collect more data.

Non-Stationary
Preferences and habits change from one day to the next.

Episodic
Pregnancy is over after nine months.
Stochastic
Cannot exactly model the DBMS's internals.

Non-Stationary
Reward of an action can change over time.

Non-Episodic
DBMS must run forever. No terminating states.
Markov Decision Process Models

Not possible to know all possible DBMS states or transition probabilities ahead of time.

Encode entire history of the DBMS in its state vector:
- Database Contents
- Database Physical Design
- Workload
- Knob Configuration
- Hardware Resources
Markov Decision Process Models

Not possible to know all possible DBMS states or transition probabilities ahead of time.

Encode entire history of the DBMS in its state vector:

- Database Contents
- Database Physical Design
- Workload
- Knob Configuration
- Hardware Resources
CREATE TABLE xxx (  
  col1 INT PRIMARY KEY,  
  col2 LONG NOT NULL,  
  col3 INT DEFAULT NULL,  
  col4 VARCHAR(8) NOT NULL
);

INSERT INTO xxx  
SELECT x.id,  
  99,  
  (RANDOM()*1000)::int,  
  SUBSTRING(MD5(id::text),1,1))  
FROM generate_series(1, 1000)  
AS x (id);
CREATE TABLE xxx (  
  col1 INT PRIMARY KEY,  
  col2 LONG NOT NULL,  
  col3 INT DEFAULT NULL,  
  col4 VARCHAR(8) NOT NULL  
);

INSERT INTO xxx  
SELECT x.id,  
  99,  
  (RANDOM() * 1000)::int,  
  SUBSTRING(MD5(id::text), 1, 1))  
FROM generate_series(1, 1000)  
AS x (id);
Database Contents State

```sql
CREATE TABLE xxx (
    col1 INT PRIMARY KEY,
    col2 LONG NOT NULL,
    col3 INT DEFAULT NULL,
    col4 VARCHAR(8) NOT NULL
);
```

Add/Drop Indexes?  
Alter Physical Layout?  
Tune Knobs?  
Change Hardware?

<table>
<thead>
<tr>
<th>Type</th>
<th>Avg Size</th>
<th>Max Size</th>
<th>Min Val</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 4, 4, 1</td>
<td>1000, 1000</td>
<td>2, 8, 8</td>
<td>99, 99</td>
<td>1,</td>
</tr>
<tr>
<td>1, 4, 4, 0</td>
<td>997, 630</td>
<td>3, 8, 1</td>
<td>'0', 'f'</td>
<td>16,</td>
</tr>
</tbody>
</table>

Table State
**Database Contents State**

```sql
CREATE TABLE xxx (
    col1 INT PRIMARY KEY,
    col2 LONG NOT NULL,
    col3 INT DEFAULT NULL,
    col4 VARCHAR(8) NOT NULL
);

ALTER TABLE xxx ADD COLUMN col5 INT;
```

**Table State**

<table>
<thead>
<tr>
<th># of Tuples</th>
<th># of Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000, 4</td>
<td></td>
</tr>
<tr>
<td>2, 8, 8</td>
<td>3, 8, 1</td>
</tr>
<tr>
<td>99, 1</td>
<td>0, 'f', 16</td>
</tr>
<tr>
<td>1000, 1000</td>
<td></td>
</tr>
<tr>
<td>97, 630</td>
<td></td>
</tr>
</tbody>
</table>

- Add/Drop Indexes?
- Alter Physical Layout?
- Tune Knobs?
- Change Hardware?
**Database Contents State**

```sql
CREATE TABLE xxx (
  col1 INT PRIMARY KEY,
  col2 LONG NOT NULL,
  col3 INT DEFAULT NULL,
  col4 VARCHAR(8) NOT NULL
);
```

<table>
<thead>
<tr>
<th># of Tuples</th>
<th># of Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000,4,</td>
<td>1,4,4,1,1000,1000,</td>
</tr>
<tr>
<td>1,4,4,1,1000,1000,</td>
<td>2,8,8,99,99,1,</td>
</tr>
<tr>
<td>1,4,4,0,997,630,</td>
<td>3,8,1,'0','f',16,</td>
</tr>
</tbody>
</table>

- **Add/Drop Indexes?**
- **Alter Physical Layout?**
- **Tune Knobs?**
- **Change Hardware?**

**ALTER TABLE xxx ADD COLUMN col5 INT;**
Database Contents State

```sql
SELECT x1.col1, COUNT(*)
FROM xxx AS x1
JOIN xxx AS x2
    ON x1.col1 > x2.col3
GROUP BY x1.col1;
```

Add Index
Database Contents State

SELECT x1.col1, COUNT(*)
FROM xxx AS x1
JOIN xxx AS x2
 ON x1.col1 > x2.col3
GROUP BY x1.col1;

Expected: 100

Estimated State

| 1000, 4, |
| 1, 4, 4, 1, 1000, 1000, |
| 2, 8, 8, 99, 99, 1, |
| 1, 4, 4, 0, 997, 630, |
| 3, 8, 2, '0', 'f', 16 |

Actual States

| 1000, 4, |
| 1, 4, 4, 1, 1000, 1000, |
| 2, 8, 8, 645, 45, 333, |
| 1, 4, 4, 34, 6188, 566, |
| 3, 8, 2, '0', 'ZZ', 100 |

| 1000, 4, |
| 1, 4, 4, 1, 1000, 1000, |
| 2, 8, 8, 111, 22, 333, |
| 1, 4, 4, 91, 18, 1985, |
| 3, 8, 2, '0', 'XXX', 33 |

⚡ Add Index
Database Contents State

```
SELECT x1.col1, COUNT(*)
FROM xxx AS x1
JOIN xxx AS x2
ON x1.col1 > x2.col3
GROUP BY x1.col1;
```

Expected: 100

Actual: -10
Database Contents State

```
SELECT x1.col1, COUNT(*)
FROM xxx AS x1
JOIN xxx AS x2
ON x1.col1 > x2.col3
GROUP BY x1.col1;
```

**Expected State:** 100

**Actual State:** -10

---

**Estimated State**

- 1000, 4,
- 1, 4, 4, 1, 1000, 1000,
- 2, 8, 8, 99, 99, 1,
- 1, 4, 4, 0, 997, 630,
- 3, 8, 2, '0', 'f', 16

---

**Actual States**

- 1000, 4,
- 1, 4, 4, 1, 1000, 1000,
- 2, 8, 8, 22, 333,
- 1, 4, 4, 0, 1985,
- 3, 8, 2, '0', 'XX', 33

---

Add Index
Database Contents State

```
SELECT x1.col1, COUNT(*)
FROM xxx AS x1
JOIN xxx AS x2
    ON x1.col1 > x2.col3
GROUP BY x1.col1;
```

**Expected:** 100

**Actual:** -10

Add Index

**Estimated State**

- 1000, 4,
- 1, 4, 4, 1, 1000, 1000,
- 2, 8, 8, 99, 99, 1,
- 1, 4, 4, 0, 997, 630,
- 3, 8, 2, '0', 'f', 16

**Actual States**

- 1000, 4,
- 1, 4, 4, 1, 1000, 1000,
- 2, 8, 8, 111, 22, 333,
- 1, 4, 4, 01, 18, 1985,
- 3, 8, 2, '0', 'XXX', 33

**ANALYZE;**
SELECT x1.col1, COUNT(*)
FROM xxx AS x1
JOIN xxx AS x2
ON x1.col1 > x2.col3
GROUP BY x1.col1;

ANALYZE;

Estimated State
1000, 4,
ários, 1, 4, 4, 1, 1000, 1000,
ários, 2, 8, 8, 99, 99, 1,
ários, 1, 4, 4, 0, 997, 630,
ários, 3, 8, 2, '0', 'f', 16

Expected: 100

Actual States
1000, 4,
ários, 1, 4, 4, 1, 1000, 1000,
ários, 2, 8, 8, 111, 22, 333,
ários, 1, 4, 4, 01, 18, 1985,
ários, 3, 8, 2, '0', 'XXX', 33

Actual: -10
Knob Configuration State

Knob Configurations

- # Exec Threads
- Parallel Query Threads
- # of Background Threads
- Buffer Pool Size
- Join Buffer Size
- Query Cache Size

16, 2, 4,...

20GB, 8GB, 500MB,...
Knob Configuration State

Knob Configurations

- # Exec Threads: 16, 2, 4, ...
- Parallel Query Threads
- # of Background Threads
- Buffer Pool Size: 20GB, 8GB, 500MB, ...
- Join Buffer Size
- Query Cache Size

32GB
Knob Configuration State

Knob Configurations

- # Exec Threads
- Parallel Query Threads
- # of Background Threads
- Buffer Pool Size
- Join Buffer Size
- Query Cache Size

32GB, 128GB
20GB, 8GB, 500MB,...
Knob Configuration State

Knob Configurations

- # Exec Threads
  - 16, 2, 4, ...

- Parallel Query Threads
  - # of Background Threads
  - 20GB, 8GB, 500MB, ...

- Buffer Pool Size
  - 32GB, 128GB

- Join Buffer Size
  - Query Cache Size

- 20GB, 8GB, 500MB, ...

- Hard Drives
  - 32GB, 128GB
Knob Configuration State

CREATE BUFFERPOOL new_bp
BLOCKSIZE 64
PAGESIZE 8192;

Knob Configurations

- # Exec Threads: 16, 2, 4, ...
- Parallel Query Threads
- # of Background Threads
- Buffer Pool Sizes: 20GB, 8GB, 500MB, ...
- Join Buffer Size
- Query Cache Size: 32GB, 128GB

32GB 128GB
Knob Configuration State

CREATE BUFFERPOOL new_bp
  BLOCKSIZE 64
  PAGESIZE 8192;

CREATE BUFFERPOOL new_bp2
  BLOCKSIZE 128
  PAGESIZE 4096;

Knob Configurations

- # Exec Threads: 16, 2, 4, ...
- Parallel Query Threads: 20GB, 8GB, 500MB, ...
- # of Background Threads: 32GB, 128GB
- Buffer Pool Sizes: 64, 8192, ...
- Join Buffer Size: 128, 4096, ...
- Query Cache Size: 32GB, 128GB
Knob Configuration State

CREATE BUFFERPOOL new_bp
  BLOCKSIZE 64
  PAGESIZE 8192;

CREATE BUFFERPOOL new_bp2
  BLOCKSIZE 128
  PAGESIZE 4096;

Knob Configurations

- # Exec Threads: 16, 2, 4, ...
- Parallel Query Threads: 20GB, 8GB, 500MB, ...
- # of Background Threads: 64, 8192, ...
- Buffer Pool Size: 64, 8192, ...
- Join Buffer Size: 128, 4096, ...
- Query Cache Size: 32GB, 128GB
Potential Solutions

1. Reduce model dimensionality to avoid variability. Examples: Hashing Kernels, Auto-Encoders, PCA. These exacerbate instability and destroy information.

2. Store hardware resource knobs as percentages relative to the available amount.

3. Use hierarchical models that isolate DBMS components to reduce the propagation of changes.
2. DATA COLLECTION

Observe the system under varying scenarios to gain experience.
Training Data Acquisition
Can only make a limited number of observations. Need to parallelize data gathering...
Seeking to Gain Experience with Pregnant Women

I have a serious problem in my life right now. My wife is pregnant and I am pretty sure that it is mine. But I don’t know anything and I can’t afford the fertility test. I have spent a lot of money on my own self and I don’t have any other option. I don’t want to lose my wife and my family. I am really worried about this. I want to study about this problem and I want to find a solution. I want to gain experience by hiring pregnant women to take care of them. I have seen that many men have done this and they have a lot of experience. I want to do this too. I am not able to collect information and I don’t understand what she is going through. In other words, I do not have the knowledge to help her and the convergence to determine the right policy for taking care of her.

Price: $100. Cash. Service supplies only.

Given my lack of good training data, I am seeking to gain more practical experience. I want to hire pregnant women to take care of them and observe their habits and preferences. This will be a couple of times a week. I want to use this experience to help me understand what she is going through. In other words, I do not have the knowledge to help her and the convergence to determine the right policy for taking care of her.

Conclusion:

I need observations. Need
Seeking to Gain Experience with Pregnant Women

I have a serious problem in my life right now. My wife is pregnant and I am pretty sure that is mine. But I don’t know anything about pregnant women. My objective is to make sure that she is happy and comfortable during this period of her life. I have been observing my wife the last two weeks but I am not able to collect information about her to understand what she is going through. I don’t know whether the actions that I am taking to help her are successful. In other words, I do not have enough data to help me converge to determine the right policy for taking care of her.

Given my lack of good training data, I am seeking to gain more experience by hiring pregnant women to hang out with. You would not have to do anything differently. I would come over to your house/apartment during the day and just observe your various habits and preferences. This will be a couple of times each week. I will bring my own food and equipment. You won’t even know that I am there. I will use the data that I collect during these observation periods to refine my models on how to make my own pregnant wife more happy.

Price: $108 cash. Serious inquiries only.

* Do NOT contact me with unsolicited services or offers.
Offline Acquisition
Targeted experiments on individual components.

Online Acquisition
Collect metrics and data from the system running in production.
Offline Training Data

End-to-End Benchmarks
Run sample workloads using the full DBMS.
Offline Training Data

1. **End-to-End Benchmarks**
   Run sample workloads using the full DBMS.

2. **Component Trainers**
   Run microbenchmarks on a subset of components.
Offline Training Data

1. End-to-End Benchmarks
   Run sample workloads using the full DBMS.

2. Component Trainers
   Run microbenchmarks on a subset of components.
Offline Collection – Logging

**End-to-End (YCSB)**

- 40,000 Observations

**Microbenchmark**

- 1,000 Observations
Online Training Data

Collect data from the DBMS running in production.

We must be more conservative about action exploration.

  Cannot allow for any perceivable effect to applications / users.
  No DBMS can fully support this today (e.g., mandatory restarts).

Use DBMS's HA replicas to explore collect training data...
Online Training Data
Online Training Data

Master

Replica

Replica

Agent

Actions

Actions

Actions
Online Training Data

Master

Replica

Failover Time?

Agent

Replica

Actions

Actions
Online Training Data

- **Master**
- **Replica**
- **Failover Time?**
- **Actions**
- **Slower**
- **Faster**

**Agent**
Online Training Data

App Server

Master

Replica

Replica

Agent
Online Training Data

App Server

Master

Replica

Replica

SQL Statements?

Physical Log?

Reads

Writes
Online Training Data

App Server ➔ Master ➔ Replica ➔ Replica ➔ App Server

- Reads
- SQL Statements?
- Physical Log?

- Reads
- Writes

Agent
Online Training Data

App Server

Master

Replica

Replica

Reads?

Reads?

SQL Statements?

SQL Statements?

Physical Log?

Physical Log?
Online Training Data
Alternative: Imitation Learning

Observe an expert tuning the DBMS and then try to train the system to mimic those decisions.

Still need to capture the state of the DBMS before each change to extrapolate why the DBA made the change.

Unlikely to work well because training data will be sparse/noisy.
3. REWARD OBSERVATION
How to identify the reward of a deployed action.
Short-Term Rewards
We immediately know when something is wrong.

Long-Term Rewards
Difficult to determine whether we are doing the correct things now for our child in the future.
Short-Term Rewards

We immediately know when something is wrong.

Long-Term Rewards

Difficult to determine whether we are doing the correct things now for our child in the future.
Short-Term Rewards
We immediately know when something is wrong.

Long-Term Rewards
Difficult to determine whether we are doing the correct things now for our child in the future.
Short-Term Rewards

We immediately know when something is wrong.

Long-Term Rewards

Difficult to determine whether we are doing the correct things now for our child in the future.

Seeking to Gain Experience with Small Children

This is an odd request, but I need to be around small children. My wife is pregnant and I don't know anything about how to take care of children. I know how to take care of my dog, but that's mostly just filling its water dish, giving it kibble, and picking up its poop. I need to know what kind of actions I should take now to ensure that my unborn child will turn out okay and not be a burden on society like I was. I would like to learn the discount sum of rewards for these actions.

As such, I am seeking to gain more experience by hiring small children to hang out with me. You can bring your son and/or daughter to my office at CMU. They will then do whatever it is that they normally do and I will try out different actions (e.g., give them candy, throw a ball at them, let them pet my dog, etc.). This will be a couple of times each week. I will bring my own food and equipment. I will use the data that I collect during these observation periods to refine my models on how to raise a small child.

I am PA Act 153 certified.

Price: $100 cash. Serious inquiries only.

- do NOT contact me with unrequested or unsolicited services or offers

Avoid scams, deal locally. Beware wire (e.g., Western Union), cashiers checks, money orders, shipping.
Short-Term Rewards

We immediately know when something is wrong.

Long-Term Rewards

Difficult to determine whether we are doing the correct things now for our child in the future.

Seeking to Gain Experience with Small Children

This is an odd request, but I need to be around small children. I am pregnant and I don’t know anything about how to take care of one. I do not know how to take care of my dog, but that’s mostly just filling his dish, giving it kibble, and picking up his poop. I need to know more about different actions I should take to ensure that my unborn child will be okay and not be a burden on society like I was. I would like to discount some of the rewards for these actions.

As such, I am seeking to gain more experience by hiring small children to hang out with me. You can bring your son and/or daughter to our home and we will take care of them. They will then do whatever it is that they normally do when they hang out with other children. We will keep a detailed record of all the actions (e.g., give them candy, throw a ball at them, pet my dog, etc.). This will be a couple of times each week. I will then pay you according to my own food and equipment. I will use the data that I collect as observation periods to reinforce my models on how to raise children.
Short-Term Rewards
Must immediately detect whether the DBMS violates constraints.

Long-Term Rewards
Identify long-term workload trends and how long it will take to prepare accordingly.
Deployment Costs

The DBMS must predict how much time and resources it will take to deploy each action.

Assume that the system only deploys one action at a time.

The cost of deploying actions can affect the objective function.
Hardware Degradations

Transient hardware problems are difficult to detect and could mask the true reward of an action.
   Likely more problematic in distributed / shared-disk systems.

Non-Elegant Solutions:
   Run periodic microbenchmarks and compare with historical data.
   Train classifiers on hardware stats (e.g., SMART).
   Tweak exploration policy to reconsider recent actions.
Reward Magnitudes

The importance of changes in the objective function depends on the workload context.
May need multiple policies to prefer one workload type over another in mixed environments.

**OLTP Workload**
- Latency: 100ms → 50ms

**OLAP Workload**
- Latency: 500ms → 250ms
Reward Magnitudes

The importance of changes in the objective function depends on the workload context.
May need multiple policies to prefer one workload type over another in mixed environments.

Mixed Workload

- Latency: 100ms → 50ms
  -50%
- Latency: 500ms → 250ms
  -50%
Reward Magnitudes

The importance of changes in the objective function depends on the workload context. May need multiple policies to prefer one workload type over another in mixed environments.
Other Hard Parts
Other Hard Parts

1. Action Selection Policies
2. Transparency / Explainability
3. Human Interaction
Weren't you building a DBMS at CMU?
Self-Driving DBMS Project

- Latch-free MVCC
- Hybrid Storage (Row + Column)
- LLVM Execution Engine
- Latch-free Bw-Tree
- Postgres Compatible
- Open-Source (Apache)

Peloton
pelotondb.io
Self-Driving DBMS Project

→ Latch-free MVCC

→ Hybrid Storage (Row + Column)

→ LLVM Execution Engine

→ Latch-free B-tree

→ Postgres Compatible

→ Open-Source (Apache)

pelotondb.io
Self-Driving DBMS Project

→ Latch-free MVCC

→ Hybrid Storage (Row + Column)

→ LLVM Execution Engine

→ Latch-free Bw Tree

→ Postgres Compatible

→ Open Source (Apache)

pelotondb.io
Peloton: Uber’s Unified Resource Scheduler for Diverse Cluster Workloads

October 30, 2018
Mountain View’s Peloton announces truck platooning partnership
Self-Driving DBMS Project
→ Latch-free MVCC
→ Hybrid Storage (Row + Column)
→ LLVM Execution Engine
→ Latch-free Bw-Tree
→ Postgres Compatible
→ Open-Source (Apache)
New System (2019)

→ Training & Modeling as First-Class Concepts
→ Zero Restart Action Deployments
→ Latch-free MVCC
→ Native Apache Arrow Storage
→ MemSQL-style LLVM Execution Engine
→ Postgres Compatible
→ Open-Source (MIT)
Peloton - NVMe-first, PostgreSQL-compatible, self-driving DBMS (pelotondb.io)

From the GitHub README:
https://github.com/cmu-db/peloton

UPDATE 2019-03-17

The Peloton project is dead. We have abandoned this repository and moved on to build a new DBMS. There are several engineering techniques and designs that we learned from this first system on how to support autonomous operations that we are doing a much better job at implementing in the second system.

We will not accept pull requests for this repository. We will also not respond to questions or problems that you may have with running this software.

Maybe this is the next DBMS they're working on?
https://github.com/cmu-db/terrier

Found a mildly amusing commit: https://github.com/cmu-db/terrier/commit/c48063978b5f3ade82...
New System (2019) →
Training & Modeling as First-
Class Concepts →
Zero Restart Action Deployments →
Latch-free MVCC →
Native Apache Arrow Storage →
MemSQL-style LLVM Execution Engine →
Postgres Compatible →
Open-Source (MIT)
Self-Driving DBMS Gym

Building a simulator is just as hard as building a real DBMS. No existing DBMS has the sufficient control APIs, metrics, and infrastructure to support rapid experimentation.

We are attempting to build the first DBMS "gym" to make it easier for ourselves and others to develop new ML methods for autonomous control.
Final Thoughts

End-to-end reinforcement learning is unlikely to work.
Will require an impossibly large amount of training data.
No existing way to model infinite action spaces.

There are engineering efforts that existing DBMSs should pursue now to support automation in the future.

I have found another way to train for fatherhood...