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

Lin Ma, William Zhang, Jie Jiao, Wuwen Wang, Matthew Butrovich, Wan Shem Lin, Prashanth Menon, Andrew Pavlo

Conor Perreault
Problem Definition

• Overall goal: help build a self-driving DBMS to reduce administration, deployment, and optimization workloads.

• Predicting runtimes and costs of different queries/jobs is one big piece of the problem.

• Many modern predictions adapt poorly to changing workloads or underlying data structures.
Similar work

- Optimizing a single query at a time using machine learning. Ignoring concurrency is unrealistic in a real system.
- Using RL to optimize pieces of query execution such as join order.
- Using RL to optimize query plan selection with systems such as Bao and Neo.
  - Neo struggles to adapt to changing workloads and schemas
  - Bao does not adapt to administrative tasks like MB2 does
Accomplishments of MB2

- MB2 breaks down the modeling problem into small tasks called operating units (OUs), which makes it easier to model complex workloads.
- MB2 models concurrent jobs within a DBMS and is able to predict interactions between the jobs.
- MB2 models administrative tasks within a database as well as query execution.
- MB2 can adapt to software updates and noisy predictions.
Operating Units

- OUs are individual pieces of a workload that are defined by developers. MB2 build a model for each OU so that a large complex workload can be predicted by adding up the pieces.
- Examples: building a hash table, garbage collection.
- Breaking down the ML problem into small pieces with OUs is similar to successful techniques in other areas, such as self-driving cars.
- Creating an OU set is work for database developers. This means they have to learn a new system, which contradicts the goal of a self-driving DBMS.
- The authors do not describe how to define an OU set well, this is left to future work.
Interference Model

- Individual OU models capture contention on locks and data structures, which is a part of concurrent contention.
- A separate interference model is designed to predict contention on hardware resources such as memory, CPU, and I/O.
- Outputs summary statistics to generalize to various concurrent OUs.
Offline Execution

- MB2 generates offline OU runners to execute different OUs and model their behavior.
- MB2 also executes concurrent runners to model interference as well.
- Offline execution is expensive since MB2 runs several different combinations of OUs. The number of combinations increases exponentially with the number of OUs.
- MB2 runs seven ML models for each OU and picks the best one based on test performance.
MB2 Architecture
Results

(a) OLAP Query Runtime Prediction

(b) OLTP Query Runtime Prediction

(a) Varying Concurrent Threads

(b) Varying Dataset Sizes
Criticism

• MB2 makes a lot of assumptions about the DBMS: In memory database and cached query execution plans
• OUs create a lot of overhead work for the database developers. Authors do not explore the optimal OU set to create.
• Cost of offline execution environment is high, and not explored in the paper.
• I think some of the ideas from MB2 are good, but there are too many limitations for it to be a useful system.
References

