Data Management By Machines For Machines

MICHAEL CAHILL, michael.cahill@sydney.edu.au, University of Sydney, Australia

As data variety, velocity, and volume explode, real-time insights remain critical and the limitations of human-centric data management systems become increasingly evident. In the near future, systems will be constructed and optimized by intelligent machines and specifically tailored for machine-based consumption. This transformation requires a radical rethinking of traditional database interfaces, interactions, and implementation.

Machine-oriented data management systems (MODMS) will leverage artificial intelligence (AI) and machine learning (ML) throughout a system's lifecycle for code generation, correctness and performance testing, autonomous configuration, data structure optimization, and workload management. A profound change lies in the design of MODMS expressly for machine consumption, which allows us to eliminate traditional human-oriented interfaces such as human-readable configuration parameters, query languages, and monitoring.

Recent work has explored the configuration space of key/value stores and demonstrated that finding optimal solutions is feasible when the input workload, budget and performance targets are known in advance. We plan to apply these ideas to the implementation of MODMS, making them easily reconfigurable so that machines can dynamically explore the configuration space and adapt to changes in workloads and budgets.

Each application using MODMS will have its own preferred data format, which will evolve over time, and MODMS should not impose an arbitrary schema on applications. However, MODMS will store data and process complex queries over structured data on behalf of multiple untrusted tenants, so executing arbitrary code supplied by clients is not safe. Thus, interfaces are still required to specify the computation over data. But since the consumers of query results are machines, they can perform further computation, opening up new opportunities to optimize where computation is performed.

A novel aspect of this approach is the potential for machine-generated code from large language models to play a significant role in the implementation of MODMS. AI-driven systems could dynamically generate optimized code for tasks such as query execution, indexing, and resource management. This raises intriguing possibilities and important considerations.

- Adaptability: MODMS would rapidly adapt to the evolving patterns and usage of the data by dynamically adjusting parameters, data layouts, and indexing, and with parts of the implementation generated on the fly as needed.
- Complexity and Verification: Ensuring the correctness and reliability of machine-generated code poses new challenges in debugging and verification.
- Completeness: Given that many open-source DBMS implementations were in the training set of LLMs, how much of a full system can they generate today?

This shift has far-reaching implications for high-performance transaction systems. By removing human bottlenecks in database management and embracing machine-oriented implementations, we will improve efficiency, adaptability, and autonomy in machine-driven deployments.

This work explores the potential of machines to build and optimize their own database management systems. We investigate the limits of code generation and dynamic reconfiguration in the context of database management systems and address the critical challenges they present.

Author's address: Michael Cahill, michael.cahill@sydney.edu.au, University of Sydney, Australia.