Using Deep Reinforcement Learning for Distributed Query Optimization

Rebecca Taft
Cockroach Labs
becca@cockroachlabs.com

ABSTRACT

Modern database systems are increasingly moving to the cloud to take advantage of economies of scale, fault tolerance, and high availability. In this new environment, traditional methods for query optimization perform suboptimally. Prior work [2] has shown that most optimizers perform poorly even on a single node when optimizing complex queries with 5 or more joins. A distributed execution environment increases the complexity several-fold. In addition to I/O and CPU utilization, distributed query optimizers must take into account data partitioning, network latency, and the possibility of network partitions and machine failures when determining the cost of a plan. The optimal plan for a single-node database may bear little resemblance to the optimal plan for a distributed database.

At Cockroach Labs, we are building a query optimizer for CockroachDB, which is an open-source, globally distributed SQL database. Our current query optimizer is a state-of-the-art cascades-style optimizer, and similar to most modern optimizers, it uses a statistics-based cost model in combination with a dynamic programming algorithm to guide the search for the best query plan. The optimizer performs well for many workloads, but for the reasons mentioned above, complex workloads on a globally distributed database may not perform well without careful manual tuning.

The goal of this project is to build a query optimizer for distributed databases such as CockroachDB that can perform well in a distributed setting and adapt to changes in the data distribution and execution environment. The fundamental challenge of building a query optimizer for a distributed database is in defining an accurate cost model that takes many constantly-changing parameters into consideration. In our approach, we avoid this issue altogther. Instead of designing a cost-model a priori, we learn it directly from the input queries and their actual execution performance.

The gist of our approach is as follows. We have a control loop system which takes a query as input, makes a sequence of transformations to the query plan based on a policy, executes the query in the cluster, and observes its execution performance to tune the policy. Specifically, we apply deep reinforcement learning (DRL) because it fits well with our control system modeling of the problem, and it can learn a complex function definition of the query transformation policy. The advantages of such a model are: (1) we avoid the error-prone process of defining the complex cost model for distributed query optimization, and (2) the model keeps learning continuously, hence adapting to the always changing workloads and data characteristics. Recent research projects [4, 3, 1] have seen promising results using reinforcement learning for single-node query optimization on a subset of relational algebra. We propose to extend this idea to distributed query optimization and apply it to the full set of queries supported by CockroachDB.

Trying to find the optimal query tree directly, given an input query, has the complication of proving logical equivalence between the output plan and the original plan, which in the general case is an unsolved problem. Instead, we aim to produce a sequence of transformations, with each transformation corresponding to an inference, and enforce that each transformation leads to a logically equivalent query plan.

This work has several challenges. First, collecting sufficient training data is expensive, because each data point involves executing a query to completion. However, we can bootstrap the process with the estimates from the existing CockroachDB query optimizer. Second, determining which queries to execute during the training period in order to most effectively learn the underlying data distribution is challenging. We can take a cue from the curriculum learning literature by carefully crafting queries that maximize the amount of information we learn from the queries and data. Therefore, despite the challenges ahead, we are confident that distributed query optimization is a tractable problem. If this proposal is accepted for CIDR 2019, we look forward to showcasing our preliminary results at the conference.

© 2018 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

1. REFERENCES

[1] S. Krishnan, Z. Yang, et al. Learning to Optimize Join Queries With Deep Reinforcement Learning. *ArXiv*, 2018.

[2] V. Leis, A. Gubichev, et al. How Good Are Query Optimizers, Really? *VLDB*, 2015.

[3] R. Marcus and O. Papaemmanouil. Deep reinforcement learning for join order enumeration. In *aiDM*, 2018.

[4] J. Ortiz, M. Balazinska, et al. Learning State Representations for Query Optimization with Deep Reinforcement Learning. In *DEEM*, 2018.