Research Proposal: Few-shot Physical Cost Estimation across different Hardware

Abstract

Query cost models are crucial for query optimization in modern DBMS. We propose a learned cost model that extends zero-shot models by generalizing across different hardware that the DBMS runs on. This approach reduces retraining costs in new environments and increases the applicability of learned models.

Problem

In modern DBMS, query cost estimation can predict the runtime for a given query plan. The estimations' accuracy plays a crucial role in query optimization and scheduling. While hand-tuned system components suffer from low accuracy and low maintainability, machine learning approaches to cost prediction have gained traction [1,2].

Cost models typically learn from runtimes of executed query plans. These training datasets are costly to compute as they require the DBMS to run thousands of queries. Hence, a significant challenge with learned cost models is **generalization across multiple *generalization dimensions*** such as database, hardware, DBMS configuration, DBMS version, or DBMS. A general model that we trained on one database could then be instantiated on an unseen database, running on a different hardware platform, without costly training data collection. **Zero-shot models** [1] represent one step towards this goal. However, their applicability is still limited, as they **only generalize across workloads and databases**.

Goal

With this work, we try to generalize zero-shot models [1] **across different hardware platforms.** The proposal contributes to the overall goal of making learned DBMS components useable out-of-the-box in unseen circumstances. The research question that we propose is:

How can we extend the Zero-shot [1] model architecture to train a cost prediction model that can perform accurately on unseen hardware platforms with little finetuning?

Solution

We propose to extend the zero-shot architecture [1] in two ways:

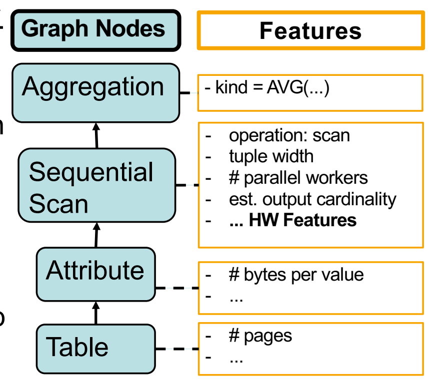
1. We can add selected **hardware features** (1) as input to the model. Open questions are: To which features will the model pay attention? Across which different platforms can the cost model generalize (e.g., is it viable to generalize across features like *CPU architecture*)?
2. We **fine-tune** the model with a small training data set on the new HW platform (few-shot learning) to improve its accuracy. Open questions: How many different hardware platforms are required to train the model? How many training examples are required on the target platform to perform accurate predictions?

HW Features (not exhaustive)	CPU arch.	# CPU cores	OS	L1 Cache size	mem. size	mem. bandwidth	mem. latency	benchmark scores

(1) HW Features (2) Data Characteristics (3) Query

(4) Transferable Query Graph (extended from [1])

The HW features can be added to the operator nodes in the query graph. Another option to explore is to only add them as input to step 7.



(5) Graph Node Encoding: one MLP per node type initializes hidden state for each node

(6) Bottom-up Message Passing: compute new hidden states as MLP of them and their children (propagation)

(7) Runtime Prediction MLP on root node

Connection to the lecture

This research proposal is based on the lecture "Learned DBMS Components" by Carsten Binnig. It extends the presented method of zero-shot learning to few-shot learning across a new generalization dimension.

Ben Hurdelhey
M. Sc. IT-Systems Engineering

Lecture Series on Database Research, WS 21/22
Hasso Plattner Institute, Potsdam, Germany

E-Mail: firstname.lastname@hpi.de

References

[1] Hilprecht, B., & Binnig, C. (2022). Zero-Shot Cost Models for Out-of-the-box Learned Cost Prediction.
[2] Sun, J., & Li, G. (2019). An End-to-End Learning-based Cost Estimator. CoRR, abs/1906.02560. http://arxiv.org/abs/1906.02560

HPI Hasso Plattner Institut