Engineering the Yannakakis Algorithm in Column Stores

Liese Bekkers

Data Science Institute, UHasselt, Belgium
liese.bekkers@uhasselt.be

Traditional binary join plans are not robust: a poorly chosen join order may compute needlessly large intermediate results, causing substantial performance degradation. Recent work by Birler, Kemper and Neumann (BKN) [1] addresses this issue by decomposing the hash join operator into two suboperators, called Lookup and Expand (L&E). By flexibly reordering these suboperations large intermediate subresults can be avoided, significantly improving query efficiency. While BKN successfully implement L&E plans inside Umbra, a compiled query engine, it is unclear, however, how to effectively implement L&E plans inside of interpreted query engines. In our ongoing work, we extend this approach to interpreted engines, with a particular focus on column stores and the processing of acyclic queries.

In this talk, we present a novel method for integrating Lookup and Expand plans in interpreted environments. We give a formal semantics to L&E plans based on the nested relational model which is an extension of the relational model where individual records may themselves contain entire relations. In particular, we design a set of nested relational operators that we call the Nested Semijoin Algebra (NSA). Here, lookup can be expressed as a form of nesting while expand is a form of unnesting. Moreover, our formalisation goes beyond lookup/expand and joins, and generalizes to all other standard relational operators. We characterize the subset of NSA plans that can be processed instance-optimally, i.e., in time asymptotically linear in input and output, and call this subset 2-phase. Two-phase NSA plans express an enumeration-based version of Yannakakis' algorithm. We also compare traditional binary join plans with NSA-based 2-phase plans in terms of a detailed cost model rather than asymptotic complexity, illustrating that when binary plans for acyclic queries can be transformed into 2-phase NSA plans by reordering unnest operations, the NSA plans cost less on all inputs. In cases where transformation is not possible, we identify conditions under which the 2-phase NSA plan is at most as costly while still being provably instance-optimal.

On established benchmarks, we show that converting existing binary join plans for acyclic queries into 2-phase NSA-based plans consistently matches or outperforms the binary plans, often achieving significant speedups while being provably instance-optimal. We hope this approach offers a fresh perspective on Yannakakis' algorithm, helping system engineers better understand its practical benefits and facilitating its adoption into a broader spectrum of query engines.

This is ongoing, unpublished research, jointly done with Frank Neven (UHasselt), Stijn Vansummeren (UHasselt), and Yisu Remy Wang (UCLA).

References

[1] A. Birler, A. Kemper, and T. Neumann. Robust join processing with diamond hardened joins. *Proc. VLDB Endow.*, 17(11):3215–3228, aug 2024.