UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

BEYOND OPTIMAL JOIN QUERIES IN GRAPH DATABASES USING THE RING

PROPUESTA DE TESIS PARA OPTAR AL TÍTULO DE MAGÍSTER EN CIENCIAS, MENCIÓN COMPUTACIÓN

Blaz Korecic Amador

Profesor guía:
Gonzalo Navarro

Santiago de Chile
Julio 2024

1. Introduction

Graph databases store data using nodes and labeled edges, emphasizing the relationships between data points. With the ever-increasing volume of data in the world, the need for efficient data processing is also growing.

Basic graph patterns are fundamental for querying graph databases, and they correspond to natural join queries over the relational representation of the graph, which is one of the most studied and important problems in database systems.

Traditionally, most database management systems have implemented pairwise join algorithms, which turn out to be suboptimal in many cases and have prompted the development of newer approaches such as multiway joins [11].

Atserias-Grohe-Marx [3] developed a method that given the sizes of input relations computes an upper bound on the size of the query result, known as the AGM bound. This led to the definition of worst-case optimality (wco): a join algorithm whose worst-case runtime meets the AGM bound [10].

The Ring [2] is a novel indexing scheme that allows worst-case optimal running time join queries and uses significantly less space than most alternatives while still having competitive experimental query times. Even though wco join algorithms meet the AGM bound given an input size, they may not be optimal in every possible instance of such queries, which creates an opportunity for improvement beyond this worst-case optimality. This thesis aims to combine the Ring with the classic Yannakakis algorithm [13] in complex cyclic queries, which would enable query faster runtimes while maintaining the space advantages the Ring offers over other alternatives.

2. Related work and concepts

The following are key concepts for understanding this thesis proposal.

2.1. AGM bound

Recently, Grohe-Marx [6] and Atserias-Grohe-Marx [3] derived an upper bound on the number of solutions in natural join queries. If Q is a query and D a database instance, the AGM bound refers to the maximum number of tuples in the query evaluation $Q(D')$ for any D' sharing the same size and structure as D.

This limits the size of the output of a join query. For the well-known triangle query:

$$Q(a, b, c) = R(a, b) \bowtie S(b, c) \bowtie T(c, a)$$

the AGM bound is $\sqrt{|R| \cdot |S| \cdot |T|}$. When $|R| = |S| = |T| = n$, we get a $n^{\frac{3}{2}}$ bound. In general, for a query Q we denote the AGM bound as Q^*.

2.2. Worst-case optimality

A join algorithm is called worst-case optimal (wco) if it has running time $\tilde{O}(Q^*)$, which is $O(Q^*)$ possibly multiplied with a polylogarithmic factor.

The NPRR algorithm [10], published in 2012, was the first wco algorithm. A popular wco algorithm is Leapfrog-Triejoin [12], which improves on the results of NPRR.

2.3. Yannakakis algorithm

The Yannakakis algorithm [13] computes the output of an acyclic join query, by creating a join tree and performing two sweeps, one bottom-up and other top-down. In each sweep, semi-joins are performed, in which each node constrains the following one. This guarantees an elimination of the tuples that don't participate in the final output, achieving a runtime of $O(|\text{input}| + |\text{output}|)$. This is the best asymptotic behavior possible, as the runtime will always be bounded by reading the input and printing the output.

2.4. EmptyHeaded

EmptyHeaded [1] is a high-level graph processing engine with a novel query optimizer. Particularly, its join queries are worst-case optimal, and in some cases outperform the AGM bound.

It computes a generalized hypertree decomposition (GHD) [4] of the query hypergraph, in which nodes are attributes and hyperedges represent relations. Each node in the GHD represents a join which is then computed with a worst-case optimal algorithm. The resulting graph is a tree, where the Yannakakis algorithm can be used to obtain the final result.

This join algorithm runs in $O(N^w + |\text{output}|)$, where w is the width of the chosen GHD, and it can be proven that it is less or equal than the exponent in the AGM bound. Finding the minimum width GHD is NP-hard [5], which is why EmptyHeaded explores by brute force all the possible GHDs to find one with minimum width. This is possible when the number of relations and attributes is small.

A problem with EmptyHeaded is that is has proven to be notoriously less space-efficient than other alternatives [2].

2.5. The Ring

The Ring [2] is an indexing scheme that supports worst-case optimal running time join queries. It uses less space than most alternatives, while still having nearly the best overall query times.

The Ring is implemented using a compact data structure called a wavelet tree [7,9], enabling the use of a Leapfrog-Triejoin variant for wco joins.

The following plot shows a comparison between the classic Ring, C-Ring (a compressed variant in exchange for runtime) and other popular indexing schemes.

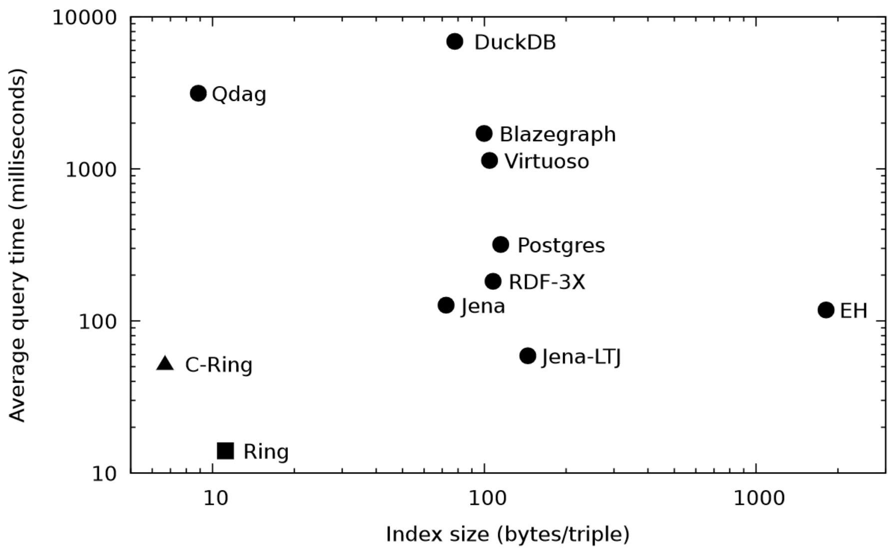

Figure 1: Index space and average query time on the Wikidata Graph Pattern Benchmark (WGPB) proposed by Hogan et al. [8]. Obtained from the original Ring article [2].

3. Problem

The Ring has proven to be slower than alternatives such as EmptyHeaded in cyclic queries [2], while greatly improving space efficiency and offering more stable times in some bad cases. This creates an opportunity of improvement which this thesis aims to develop, improving the runtime of the Ring in these cyclic queries while maintaining its other advantages.

The proposed strategy is to use some method to decompose the query graph into cycles, solve these cycles using the Ring and in the remaining acyclic graph use the Yannakakis algorithm. Multiple decomposition methods may be tested, such as finding the biconnected components or finding a minimum width GHD, similarly to what EmptyHeaded does.

4. Research questions

- What method of decomposing the query graph offers better performance?
- Does this approach offer better runtime compared to the previous approach with the Ring?
- Does this approach offer better space and time trade-offs compared to other modern indexing schemes such as EmptyHeaded?

5. Hypothesis

This approach of decomposing a query graph in cycles, solving the cycles using the Ring and then computing the final answer in the resulting acyclic query using the Yannakakis algorithm offers better runtime than the previous approach of solving the entire query using only the Ring and better space efficiency than alternatives such as EmptyHeaded.

6. Objectives

6.1. General objective

Advance the state-of-the-art on solving cyclic join queries using the Ring, beyond the asymptotic worst-case optimal behavior it already presents.

6.2. Specific objectives

- Design, implement and test an implementation of the complete algorithm.
- Benchmark and compare this implementation with other approaches to solving complex join queries.

7. Methodology

The following is a preliminary sequence of steps to complete this work.

1. Study the source code of the Ring implementation in `C++`.
2. Simple implementation with the Ring that takes an already decomposed graph into a tree where each node is a sub-query in the original graph, containing the cycles and solves those sub-queries.
3. Evaluate implementation alternatives of the Yannakakis algorithm.
4. Connect the previous algorithm with the Yannakakis algorithm to obtain a final output.
5. Study and evaluate alternatives for the graph decomposition, such as decomposing into biconnected components or finding a minimum width GHD.
6. Implement a graph decomposition algorithm and connect it with the rest of the previously developed algorithm.
7. Test and benchmark all the implementations.
8. Compare with other results, most importantly the original Ring results and EmptyHeaded.

All the implementations will be written in `C++`, for its efficiency, familiarity and ease of integration with already implemented algorithms such as the Ring.

8. Expected results

The expected result of this work is a beyond wco optimal join algorithm that, when applied to a cyclic query graph:

- Improves the runtime of the Ring, and
- offers similar or better runtimes to EmptyHeaded with much better space efficiency.

This work will be materialized in a publishable thesis and a publicly available, open-source implementation that allows for further experimentation by other authors.

Bibliography

[1] Christopher R. Aberger, Andrew Lamb, Susan Tu, Andres Nötzli, Kunle Olukotun, and Christopher Ré. 2017. EmptyHeaded: A Relational Engine for Graph Processing. *ACM Trans. Database Syst.* 42, 4 (October 2017). https://doi.org/10.1145/3129246

[2] Diego Arroyuelo, Adrián Gómez-Brandón, Aidan Hogan, Gonzalo Navarro, Juan Reutter, Javiel Rojas-Ledesma, and Adrián Soto. 2024. The Ring: Worst-case Optimal Joins in Graph Databases using (Almost) No Extra Space. *ACM Trans. Database Syst.* 49, 2 (March 2024). https://doi.org/10.1145/3644824

[3] Albert Atserias, Martin Grohe, and Dániel Marx. 2008. Size Bounds and Query Plans for Relational Joins. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008. 739–748. https://doi.org/10.1109/FOCS.2008.43

[4] Georg Gottlob, Martin Grohe, nysret Musliu, Marko Samer, and Francesco Scarcello. 2005. Hypertree decompositions: structure, algorithms, and applications. In *Proceedings of the 31st International Conference on Graph-Theoretic Concepts in Computer Science (WG'05)*, 2005. Springer-Verlag, Metz, France, 1–15. https://doi.org/10.1007/11604686_1

[5] Georg Gottlob, Zoltán Miklós, and Thomas Schwentick. 2009. Generalized hypertree decompositions: NP-hardness and tractable variants. *J. ACM* 56, 6 (September 2009). https://doi.org/10.1145/1568318.1568320

[6] Martin Grohe and Dániel Marx. 2014. Constraint Solving via Fractional Edge Covers. *ACM Trans. Algorithms* 11, 1 (August 2014). https://doi.org/10.1145/2636918

[7] Roberto Grossi, Ankur Gupta, and Jeffrey Scott Vitter. 2003. High-order entropy-compressed text indexes. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '03)*, 2003. Society for Industrial, Applied Mathematics, Baltimore, Maryland, 841–850.

[8] Aidan Hogan, Cristian Riveros, Carlos Rojas, and Adrián Soto. 2019. A Worst-Case Optimal Join Algorithm for SPARQL. In *The Semantic Web – ISWC 2019*, 2019. Springer International Publishing, Cham, 258–275.

[9] Gonzalo Navarro. 2014. Wavelet trees for all. *Journal of Discrete Algorithms* 25, (2014), 2–20. https://doi.org/https://doi.org/10.1016/j.jda.2013.07.004

[10] Hung Q. Ngo, Ely Porat, Christopher Ré, and Atri Rudra. 2012. Worst-case Optimal Join Algorithms. Retrieved from https://arxiv.org/abs/1203.1952

[11] Hung Q. Ngo, Christopher Re, and Atri Rudra. 2013. Skew Strikes Back: New Developments in the Theory of Join Algorithms. Retrieved from https://arxiv.org/abs/1310.3314

[12] Todd L. Veldhuizen. 2013. Leapfrog Triejoin: a worst-case optimal join algorithm. Retrieved from https://arxiv.org/abs/1210.0481

[13] Mihalis Yannakakis. 1981. Algorithms for acyclic database schemes. In *Proceedings of the Seventh International Conference on Very Large Data Bases - Volume 7 (VLDB '81)*, 1981. VLDB Endowment, Cannes, France, 82–94.