An Efficient GPU-based Path Querying of the Graph Databases

Rustam Azimov

Keywords

CFPQ, context-free path querying, graph databases, linear algebra, matrix operations, GPGPU, sparse matrices, formal language theory, context-free grammars.

1 Introduction

Graphs are used as a data structure to represent large volumes of information in a compact and convenient for analysis form in many areas: graph databases [1, 2, 3], bioinformatics [4, 5], static code analysis [6, 7], etc. In these areas, it is necessary to evaluate queries for large graphs in order to determine the dependencies between the nodes.

The most popular queries are path queries which use regular or context-free grammars for constraints on the path. The use of context-free grammars instead of regular grammars in path querying allows us to formulate more complex queries to the graph and solve a wider class of problems.

There is a number of algorithms for context-free path query evaluation (CFPQ) [5, 8, 9, 10]. However, these algorithms demonstrate a poor performance when applied to large graphs.

One of the most popular techniques used to increase performance when working with large data is the use of a GPU to perform computation. We proposed the only known approach in this area that makes it possible to use the GPUs efficiently [11]. This is the matrix approach, in which the adjacency matrix of the input graph is built, the elements of which are sets of non-terminals of the input grammar. Next, the transitive closure of the constructed matrix is calculated using the derivation rules of the input grammar. In the process of computing, the operations of multiplication and addition of Boolean matrices are actively used and can be computed very efficiently using the GPUs. For example, such libraries as cuBLAS, cuSPARSE, or CUTLASS provide high-performance implementations of necessary matrix operations (GEMM, etc.). We are especially interested in the CUTLASS library because it provides the ability to customize matrix types. We already have some implementations [12] which show that GPGPU utilization for Boolean matrices multiplication can significantly increase the performance of CFPQs evaluation.

The project is devoted to the study of our new matrix-based path querying approach and creating a high-performance implementations for analyzing large graphs. The active use of matrix

operations in our approach makes it possible to efficiently apply a wide class of matrix optimizations and computing techniques (GPGPU, parallel processing, sparse matrix representation, distributed-memory computation, etc.).

2 Motivation

The context-free path querying can be very useful when applied to the graph databases [3] and there are projects which are interested in a high-performance context-free path querying implementations.

There is the GraphBLAS that provides standardized building blocks of graph algorithms, optimizes and simplifies computation of many different graph queries. We plan to provide a high-performance context-free path querying implementations as part of GraphBLAS. In these implementations we will use sparse matrix representation and with the help of Nvidia libraries we will apply such optimizations as GPGPU and distributed-memory computation.

Within it, there is the RedisGraph graph database which uses both GraphBLAS and Cypher query language. Therefore, we also plan to contribute a high-performance implementations of the expressive graph database queries to the RedisGraph. RedisGraph already uses linear algebra blocks from GraphBLAS, however, providing a GPU implementations for these blocks will significantly speed up graph database query evaluation. Finally, our success can be a good motivation to extend Cypher, the industry's most widely adopted graph database query language, with context-free paths constraints.

References

[1] RedisGraph. `https://redislabs.com/redis-enterprise/redis-modules/redis-enterprise-modules/redisgraph/`. [Online; accessed 10-September-2019].

[2] A. Mendelzon and P. Wood. Finding regular simple paths in graph databases. *SIAM J. Computing*, 24(6):1235–1258, 1995.

[3] Jochem Kuijpers, George Fletcher, Nikolay Yakovets, and Tobias Lindaaker. An experimental study of context-free path query evaluation methods. In *Proceedings of the 31st International Conference on Scientific and Statistical Database Management*, pages 121–132. ACM, 2019.

[4] Christian Theil Have and Lars Juhl Jensen. Are graph databases ready for bioinformatics? *Bioinformatics*, 29(24):3107, 2013.

[5] Petteri Sevon and Lauri Eronen. Subgraph queries by context-free grammars. *Journal of Integrative Bioinformatics*, 5(2):100, 2008.

[6] John Kodumal and Alex Aiken. The set constraint/cfl reachability connection in practice. *ACM Sigplan Notices*, 39(6):207–218, 2004.

[7] Qirun Zhang, Michael R Lyu, Hao Yuan, and Zhendong Su. Fast algorithms for dyck-cfl-reachability with applications to alias analysis. In *ACM SIGPLAN Notices*, volume 48, pages 435–446. ACM, 2013.

[8] Semyon Grigorev and Anastasiya Ragozina. Context-free path querying with structural representation of result. *arXiv preprint arXiv:1612.08872*, 2016.

[9] Jelle Hellings. Conjunctive context-free path queries. In *Proceedings of ICDT'14*, pages 119–130, 2014.

[10] X. Zhang, Z. Feng, X. Wang, G. Rao, and W. Wu. Context-free path queries on rdf graphs. In *International Semantic Web Conference*, pages 632–648. Springer, 2016.

[11] Rustam Azimov and Semyon Grigorev. Context-free path querying by matrix multiplication. In *Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, page 5. ACM, 2018.

[12] Nikita Mishin, Iaroslav Sokolov, Egor Spirin, Vladimir Kutuev, Egor Nemchinov, Sergey Gorbatyuk, and Semyon Grigorev. Evaluation of the context-free path querying algorithm based on matrix multiplication. In *Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, page 12. ACM, 2019.