Graph-Scrutinizer: Towards Massive Graph Analytics and Reasoning

Jože Rožanec
Jožef Stefan Institute
Slovenia
joze.rozanec@ijs.si

Michael Cochez
Vrije Universiteit Amsterdam
The Netherlands
m.cochez@vu.nl

Ruud van Bakel
Vrije Universiteit Amsterdam
The Netherlands
r.van.bakel@vu.nl

Brian Elvesæter
SINTEF
Norway
brian.elvesater@sintef.no

Dumitru Roman
SINTEF
Norway
dumitru.roman@sintef.no

ABSTRACT

Graphs can represent various phenomena and are increasingly used to tackle complex problems. Among the challenges associated with graph processing is the ability to analyze and mine massive-scale graphs. While the massive scale is usually associated with distributed systems, the complex nature of graphs makes them an exception to the rule. Currently, most graph processing is performed within a single computer. In this research, we describe a solution at a conceptual level in the context of the Graph-Massivizer architecture. We use two approaches to provide graph analytics and querying functionalities at scale. First, we leverage graph sampling techniques to obtain relevant samples and avoid processing the whole graph. Second, we support heuristic and neural query execution engines. We envision an interface that will decide which queries to execute with a given engine, given constraints (e.g., execution time boundaries, exactness of results, energy saving requirements).

CCS CONCEPTS

• **Computer systems organization → Architectures**.

ACM Reference Format:
Jože Rožanec, Michael Cochez, Ruud van Bakel, Brian Elvesæter, and Dumitru Roman. 2023. Graph-Scrutinizer: Towards Massive Graph Analytics and Reasoning. In *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering (ICPE '23 Companion), April 15–19, 2023, Coimbra, Portugal.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3578245.3585338

1 INTRODUCTION

Graphs are mathematical abstractions used to model relationships between objects. They represent real-world phenomena and allow us to explore, predict and explain such phenomena [13]. Among prevalent domains modeled with graphs, we find social networks, the world wide web, telecommunications, fraud detection, logistics, and epidemiology [3, 12]. Growing attention and research are devoted to graph processing [4, 7, 8]. The information in graph-based data can be retrieved by exploring the topology and attribute information. In this position paper, we describe how we envision the *Graph-Scrutinizer*, a novel graph processing component defined within the Graph-Massivizer architecture (see Fig. 2), which aims to enable flexible and efficient graph processing at scale, even when the graph changes over time [11].

2 BACKGROUND

Graphs are usually persisted and processed in single-machine systems, distributed systems, or high-performance computing clusters. Proper graph storage plays a fundamental role in enabling efficient graph processing. In the project, the storage of graphs is in two parts. The full graph is stored in a graph database by the Graph Inceptor, and the Graph-Scrutinizer will maintain specific details of the information to speed up its ability to execute its tasks. Some of this will be stored in graph format, some in a neural form (e.g., weights of machine learning models), or other structures. We find persistence data formats and partitioning strategies in related work on storing large graphs. Data formats exploit general graph characteristics [2] or domain-specific knowledge [1] to enable certain compression levels. Similarly, it is possible to use compression to scale the capabilities of machine learning models which work with graph data [6]. Graph partitioning divides the graph over multiple storages while minimizing the number of edges between these subgraphs. The chosen partition strategies must be aligned with the graph processing approaches.

Dominguez-Sal et al. [5] categorized graph processing algorithms into seven categories: traversals, graph analysis, component identification, communities detection, centrality measure computation, pattern matching, and graph anonymization. All of these are of relevance for the Graph-Scrutinizer. Lumsdaine et al. [9] consider the most significant challenges in graph processing identifying the elementary unit of computation, dealing with the poor data locality, and assigning computation units given the skewed nature of graphs and the high data-access-to-computation ratio. Usually, graph processing algorithms that execute over the whole graph differ fundamentally from those solved with graph query languages. Nevertheless, new approaches are being proposed, which blur the requirement lines between both. For example, neural approaches have recently been proposed to efficiently answer complex queries on knowledge graphs [10].

ICPE '23 Companion, April 15–19, 2023, Coimbra, Portugal

ACM ISBN 979-8-4007-0072-9/23/04.
https://doi.org/10.1145/3578245.3585338

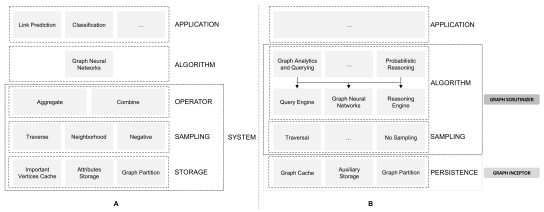

Figure 1: Parallelism between AliGraph architecture and Graph-Scrutinizer component of the Graph-Massivizer architecture. (A) depicts the AliGraph [14] architecture, while (B) describes a possible realization of the Graph-Scrutinizer and Graph-Inceptor in the context of the Graph-Massivizer architecture.

3 GRAPH-SCRUTINIZER

The Graph-Scrutinizer has some overlap with the AliGraph architecture [14] (see Fig. 1). While the AliGraph architecture was designed to support graph neural networks at scale only, we generalize the design to support graphs either persisted in a single instance or distributed setting while providing a more comprehensive range of analytic and querying capabilities. We consider two building blocks: samplers and algorithms. The samplers enable us to sample relevant data from the graph, considering some appropriate criteria. Conversely, the algorithms provide means for performing graph queries, analytics, and probabilistic reasoning.

The Graph-Massivizer revolves around the timely delivery of results and sustainability. Therefore, we balance time and space complexity and consider infrastructure capacity limits and awareness of energy requirements and their environmental impact. In some cases, an approximation of the correct answer arriving in time with low energy consumption is better than a correct answer arriving too late or consuming a lot of energy. To do this, there are multiple algorithms with different characteristics for each task. Dynamically, the choices are optimized depending on the current requirements. For example, we plan to support two implementations for querying and reasoning: one using heuristics and a neural one. The heuristic one can retrieve exact results, but is slower, while the neural one would execute faster but not provide correctness guarantees.

Graph-Scrutinizer will also have an API, providing a standard interface for query and reasoning requests, which aims to abstract the client from the particularities of the underlying implementation and constraints. Given the tradeoff between the heuristic and neural approaches, we envision having a component to weigh and decide which approach to use based on the characteristics of the request, infrastructure, runtime configurations, and execution optimization goals (e.g., energy saving, and precision levels, among others). These decisions would be enabled by the graph optimizer and greenifier components.

ACKNOWLEDGMENT

The Graph-Massivizer project receives funding from the Horizon Europe research and innovation program of the European Union under grant agreement 101093202. The project started on January 1st, 2023, and lasts three years.

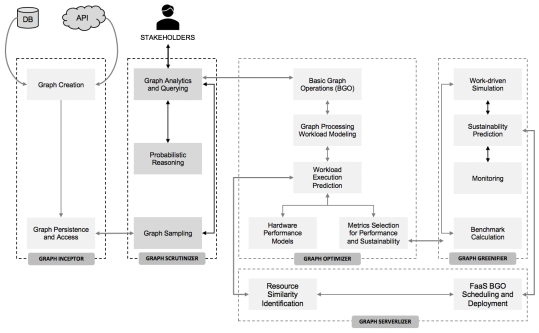

Figure 2: The Graph-Massivizer technical architecture, highlighting the Graph-Scrutinizer component. [11]

REFERENCES

[1] P. Boldi and S. Vigna. 2004. The Webgraph Framework I: Compression Techniques. In *Proceedings of the 13th International Conference on World Wide Web* (New York, NY, USA) *(WWW '04)*. Association for Computing Machinery, New York, NY, USA, 595–602. https://doi.org/10.1145/988672.988752

[2] Miguel E Coimbra, Sérgio Esteves, Alexandre P Francisco, and Luís Veiga. 2022. VeilGraph: incremental graph stream processing. *Journal of Big Data* 9, 1 (2022), 23.

[3] Miguel E Coimbra, Alexandre P Francisco, and Luís Veiga. 2021. An analysis of the graph processing landscape. *journal of Big Data* 8, 1 (2021), 1–41.

[4] Laxman Dhulipala, Guy E. Blelloch, and Julian Shun. 2019. Low-Latency Graph Streaming Using Compressed Purely-Functional Trees. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation* (Phoenix, AZ, USA) *(PLDI 2019)*. Association for Computing Machinery, New York, NY, USA, 918–934. https://doi.org/10.1145/3314221.3314598

[5] David Dominguez-Sal, Norbert Martinez-Bazan, Victor Muntes-Mulero, Pere Baleta, and Josep Lluis Larriba-Pey. 2011. A Discussion on the Design of Graph Database Benchmarks. In *Performance Evaluation, Measurement and Characterization of Complex Systems*, Raghunath Nambiar and Meikel Poess (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 25–40.

[6] Alessandro Generale, Till Blume, and Michael Cochez. 2022. Scaling R-GCN Training with Graph Summarization. In *Companion Proceedings of the Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 1073–1082. International Workshop on Graph Learning at the Webconf 2022 arXiv preprint arXiv:2203.02622.

[7] Raehyun Kim, Chan Ho So, Minbyul Jeong, Sanghoon Lee, Jinkyu Kim, and Jaewoo Kang. 2019. HATS: A Hierarchical Graph Attention Network for Stock Movement Prediction. https://doi.org/10.48550/ARXIV.1908.07999

[8] Sandra Maria Correia Loureiro, João Guerreiro, and Iis Tussyadiah. 2021. Artificial intelligence in business: State of the art and future research agenda. *Journal of business research* 129 (2021), 911–926.

[9] Andrew Lumsdaine, Douglas Gregor, Bruce Hendrickson, and Jonathan Berry. 2007. Challenges in parallel graph processing. *Parallel Processing Letters* 17, 01 (2007), 5–20.

[10] Pasquale Minervini, Erik Arakelyan, Daniel Daza, and Michael Cochez. 2021. Complex Query Answering with Neural Link Predictors. In *International Conference on Learning Representations 2021*. openreview.net, online, 1–14.

[11] Radu Prodan, Dragi Kimovski, Andrea Bartolini, Michael Cochez, Alexandru Iosup, Evgeny Kharlamov, Jože Rožanec, Laurenţiu Vasiliu, and Ana Lucia Vărbănescu. 2022. Towards Extreme and Sustainable Graph Processing for Urgent Societal Challenges in Europe. In *2022 IEEE Cloud Summit*. IEEE, 23–30.

[12] Siddhartha Sahu, Amine Mhedhbi, Semih Salihoglu, Jimmy Lin, and M Tamer Özsu. 2020. The ubiquity of large graphs and surprising challenges of graph processing: extended survey. *The VLDB journal* 29 (2020), 595–618.

[13] Sherif Sakr, Angela Bonifati, Hannes Voigt, Alexandru Iosup, Khaled Ammar, Renzo Angles, Walid Aref, Marcelo Arenas, Maciej Besta, Peter A Boncz, et al. 2021. The future is big graphs: a community view on graph processing systems. *Commun. ACM* 64, 9 (2021), 62–71.

[14] Hongxia Yang. 2019. Aligraph: A comprehensive graph neural network platform. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 3165–3166.