Scalable Analytics on Multi-Streams Dynamic Graphs

Muhammad Ghufran Khan

October 24, 2022

Scalable Analytics on Multi-Streams Dynamic Graphs

Muhammad Ghufran Khan
Inria & Institut Polytechnique de Paris
Palaiseau, France
muhammad.khan@inria.fr

ABSTRACT

Several real-time applications rely on dynamic graphs to model and store data arriving from multiple streams. In addition to the high ingestion rate, the storage and query execution challenges are amplified in contexts where consistency should be considered when storing and querying the data. This Ph.D. thesis addresses the challenges associated with multi-stream dynamic graph analytics. We propose a database design that can provide scalable storage and indexing, to support consistent read-only analytical queries (present and historical), in the presence of real-time dynamic graph updates that arrive continuously from multiple streams.

KEYWORDS

Dynamic graph, read-only present and historical queries, multi-stream graph processing

1 INTRODUCTION

Dynamic graphs are omnipresent in the context of real-time applications that generate massive amounts of events. These events can be seen as high-velocity data stream, whose timely analysis is critical for applications such as monitoring of cyber attacks in system security applications [10], fraud detection in financial institutions [10], anomaly detection in computer networks [3] and many more.

This PhD thesis considers the above-mentioned real-time use cases. We consider a database modeled as a labeled property graph (LPG), that is continuously updated. Updates may arrive from multiple streams, creating a need for appropriate transaction support, in order to avoid inconsistencies; on such graph, we must be able to answer analytical queries about the current graph, as well as historical queries. In this short paper, we present the challenges (Section 2), outline the approach we investigate (Section 3) and finally related work (Section 4) before concluding (Section 5).

2 CHALLENGES

In our problem setting, we focus on systems that can continuously ingest temporal labeled property graph (LPG) updates from multiple streams while at the same time supporting a diverse set graph analytic queries with clear consistency guarantees. We consider complex scenarios where graph updates are generated from multiple streams; updates from a given stream may arrive later due to any kind of transmission delay, leading to *out-of-order* [1] updates, which may lead to consistency issues. Figure 1 exemplifies a sequence of updates labeled A to E, considering the update generation time (the time when the update is generated from the source machine) V_T, with update B arriving out of order (after D). After

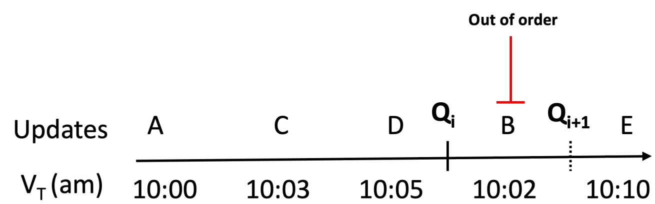

Figure 1: Example scenario for out-of-order update

the first three *received* updates have been ingested, that is, reflected on the database state, query Q arrives and requires two iterations on the updates ingested up to that time. In the first iteration, the query Q_i computes on the updates (A, C, and D), and before starting the second iteration $Q_{(i+1)}$, the delayed update B arrives and may thus be reflected in the computed query results. This may lead to an inconsistency, since B was not reflected in the first part of Q's evaluation.

There are several challenges associated with multi-stream dynamic graph analytics listed below:

- **C1**: How to enable data ingestion from multiple streams and diverse set of consistent graph analytics in parallel?
- **C2**: How to efficiently query the present and historical graph updates of the system?
- **C3**: How to store both topological and properties updates of the graph?
- **C4**: How to provide consistency guarantee in presence of out-of-order [1] updates?

Several works [2, 3, 5, 7, 10] have been published in the past to address partially the challenges described above; we discuss them in detail in Section 4.

3 APPROACH

The objective of this PhD thesis is to devise a scalable system that can answer analytical queries (present, historical) over dynamic graphs, updated multiple streams, whose updates may arrive out-of-order.

The strategy we take in my PhD project to address the challenges listed in Section 2 is as follows:

- We maintain a *write data store*, which is the recipient of all write operations. It supports a multi-version concurrency control (MVCC) [9] protocol, in which temporal graph updates from multiple streams are considered *write-only* transactions, and analytical queries (present and historical) are executed as *read-only* transactions. This ensures consistent graph analytic even in the presence of out-of-order updates, addressing challenges **C1** and **C4**.
- We explore time-based indexing techniques [4, 6] in order to design an effcient *read store*, the default recipient of all

BDA 2022, October 24–27, 2022, Clermont-Ferrand, France
2022. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

State-of-the-art	Present/History query	Topological/Attribute	Out-of-order update
Llama	YES/NO	YES/YES	NO
GraphOne	YES/NO	YES/NO	NO
LiveGraph	YES/NO	YES/YES	NO
Teseo	YES/NO	YES/NO	NO
Tegra	YES/YES	YES/YES	NO
This Thesis	YES/YES	YES/YES	**YES**

Table 1: Comparison of state-of-the-art systems in the context of our dynamic graph challenges.

read-only transactions, on top of the write store. The read store allows efficient processing of analytical point and interval queries (both present and historical), thus addressing challenge **C2**. Data brought by updates in the write-store is periodically moved to the read-store to make it available for queries.

- To address challenge **C3**, we design the system in such a way that the storage of the topological graph updates is decoupled from the storage of graph properties updates. Keeping them separate allows us to optimize data access paths separately for these two largely orthogonal components of the property graph.

Currently, we are in a design phase, considering the data structures that can best fit our system requirements.

4 RELATED WORK

Existing systems that support multi-stream dynamic graph analytics are classified into two main categories: a) the systems that compute continuous results of pre-registered, fixed queries, such as path queries, over dynamic graphs; an example is [8]; b) systems that enable the evaluation of ad-hoc analytic queries on top of dynamic graphs; these include Llama [7], Teseo [5], LiveGraph [10], Tegra [2]. GraphOne [3] appears as a hybrid between the two, as it provides a hybrid store that supports both real-time and batch analytic queries.

Our work is closely related to the group b), and Table 1 summarizes the challenges associated with these systems. Llama [7] is a graph analytics system using multi-version arrays that support snapshot isolation. Other systems that provide snapshot isolation are GraphOne [3] and Tegra [2]. GraphOne utilizes a multi-version degree array and Tegra uses persistent adaptive radix trees (PART) to maintain different snapshot versions. Mission critical applications, such as banking applications, need transactional guarantees while ingesting updates in dynamic graph; Teseo [5] and LiveGraph [10] were designed to address such requirements. The fat tree (B+ tree together with a sparse array) data structure is used in Teseo to ingest structural updates to the graph; LiveGraph [10] uses a transaction edge log and vertex blocks to support updates to labeled properties graph. Both systems use multi-version concurrency control (MVCC) [9] protocol to maintain data consistency.

All the systems mentioned in Table 1, as well as the one we aim for, support queries based on the current state of the dynamic graph. Except for Tegra [2] and this thesis, no other system supports historical state queries. This thesis, LiveGraph [10], Llama [7], and Tegra [2], support both topological and attribute updates. Existing systems do not provide consistency guarantees in the presence of out-of-order updates, while we aim to attain this goal as explained above. A position paper appeared [1] on the topic of bitemporal dynamic graph analytics, that addresses all the challenges that are described in Table 1, as we do. However, there is no follow-up published.

5 CONCLUSION AND PERSPECTIVES

This PHD thesis aims to provide a scalable storage and indexing solution that can ingest real-time graph updates from multiple streams and, on top of that, provide consistent graph analytics (read-only present and historical queries) even in the presence of out-of-order updates. In the near future, we should finalize the choice of data structures for our system design, and implement it to provide a comprehensive proof-of-concept system.

Thesis context

My PhD started in October 2021, and my PhD advisers are Ioana Manolescu and Angelos Anadiotis.

Acknowledgments. This work is funded by AI Chair SourcesSay project (ANR-20-CHIA-0015) grants.

REFERENCES

[1] Hassan Halawa and Matei Ripeanu. 2021. Position Paper: Bitemporal Dynamic Graph Analytics *(GRADES-NDA '21)*.

[2] Anand Padmanabha Iyer, Qifan Pu, Kishan Patel, Joseph E. Gonzalez, and Ion Stoica. 2021. TEGRA: Efficient Ad-Hoc Analytics on Evolving Graphs. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. USENIX Association, 337–355.

[3] Pradeep Kumar and H. Howie Huang. 2019. GraphOne: A Data Store for Real-time Analytics on Evolving Graphs. In *17th USENIX Conference on File and Storage Technologies (FAST 19)*. USENIX Association, Boston, MA, 249–263.

[4] Viktor Leis, Alfons Kemper, and Thomas Neumann. 2013. The adaptive radix tree: ARTful indexing for main-memory databases. In *2013 ICDE*. 38–49.

[5] Dean De Leo and Peter A. Boncz. 2021. Teseo and the Analysis of Structural Dynamic Graphs. *Proc. VLDB Endow.* 14 (2021), 1053–1066.

[6] Siqiang Luo, Subarna Chatterjee, Rafael Ketsetsidis, Niv Dayan, Wilson Qin, and Stratos Idreos. 2020. Rosetta: A Robust Space-Time Optimized Range Filter for Key-Value Stores. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (2020).

[7] Peter Macko, Virendra J. Marathe, Daniel W. Margo, and Margo I. Seltzer. 2015. LLAMA: Efficient graph analytics using Large Multiversioned Arrays. In *2015 IEEE 31st International Conference on Data Engineering*. 363–374.

[8] Anil Pacaci, Angela Bonifati, and M. Tamer Özsu. 2022. Evaluating Complex Queries on Streaming Graphs. *In Proceedings of the 38th IEEE International Conference on Data Engineering (ICDE)* (2022).

[9] Yingjun Wu, Joy Arulraj, Jiexi Lin, Ran Xian, and Andrew Pavlo. 2017. An empirical evaluation of in-memory multi-version concurrency control. *Proceedings of the VLDB Endowment* 10 (03 2017), 781–792.

[10] Xiaowei Zhu, Guanyu Feng, Marco Serafini, Xiaosong Ma, Jiping Yu, Lei Xie, Ashraf Aboulnaga, and Wenguang Chen. 2020. LiveGraph: a transactional graph storage system with purely sequential adjacency list scans. *Proceedings of the VLDB Endowment* 13 (03 2020), 1020–1034.