EPITOME: Summarization and Structuring of Continuous Data in Concurrent Processing Pipelines

DEBS 2025 Doctoral Symposium

Martin Hilgendorf
Supervised by Marina Papatriantafilou
Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
Gothenburg, Sweden
martin.hilgendorf@chalmers.se

Keywords

Data pipelines, concurrency, data summarization, data sketches

1 Problem Statement

The core problem investigated in my PhD is about exploring and developing summarization techniques for continuous data, with a particular focus on concurrency to support high-rate streams. *Data Sketches* are a common class of data summarization algorithms, and have been studied for several decades. They probabilistically reduce large amounts of input data to significantly smaller summaries that can be queried for certain statistical aspects of the input, depending on the type of sketch. As data rates and volumes continue to increase, more efficient processing and analysis techniques become necessary. Data sketches provide solutions to many of the involved challenges: featuring efficient operations enables them to keep up with high-rate streams, while a small (oftentimes constant) memory footprint supports . Due to this, sketches have found widespread use in common data analytics tools, including the Apache DataSketches library [2] and Redis [9].

My current work investigates parallelizing operations on sketches; the challenges involved are many. For example, what *semantics* will a query have that executes concurrently with updates — will it observe the concurrent updates? Will it observe them partially, perhaps producing inconsistent results? Will it ignore them entirely, producing stale results? What about updates completed before the query?

Simultaneously, we are investigating sketches that can provide answers to multiple queries from the same data structure. Typically, each query would require a dedicated sketch. However, this causes prohibitive overheads as well as consistency concerns — the different sketches might have processed different amounts of input, and accordingly, their query results will reflect different "views of the world". Thus, it seems useful to investigate the possibilities for estimating multiple kinds of queries from a single sketch data structure.

Other problems on the radar include summarization techniques other than sketches, such as wavelets, and summaries for other types of data, e.g. graphs.

2 Research approach/Methodology

The research methodology includes the design and analysis of concurrent and/or summarization algorithms, and implementing them. The implementations are used for detailed empirical evaluations and benchmarks, exploring sensitivity of the solutions to many variables, e.g. thread counts, memory, parameters of the data structure and algorithm, input data characteristics (rate, skewness), etc.

3 Research setting

Research questions for my current work include:

- Which challenges are involved in designing sketches for multiple queries and concurrent operations?
- Which synchronization design is "sufficiently accurate"?
- Which consistency semantics do we have across queries?

4 State of the art / Related work

Data sketches have been studied for decades, with the seminal AMS sketch [3] appearing as early as 1999. The well-known Count-Min Sketch [7] followed in 2005, by Graham Cormode who continues to be active in the area, e.g. with the recent *Applications of Sketching and Pathways to Impact* [5].

A co-author of Cormode is Minos Garofalakis, whose work on sketches include the Fast-AGMS (2005) [6] for estimating F_2, and recently followed by OmniSketch [10] for enabling analysis of multidimensional streams using sketches and streams.

Works on concurrent sketches include SKT [4] and the Delegation Sketch [12]. While these works target concurrent operations (SKT parallelizes updates but not queries, Delegation Sketch parallelizes updates with point queries), we have investigated sketches that answer multiple queries, concurrently with updates.

Works on synchronization and concurrency semantics that have been utilized in my current work include [8] which introduces weak regularity, and [11], which formalizes Intermediate Value Linearizability, a powerful and useful consistency notion for concurrent objects, allowing to relax strict semantics for high efficiency gains.

5 Research approach

We have developed a partitioned data structure, based on the delegation design, which allows threads to operate concurrently with minimal communication overhead. To enable concurrent queries, we have designed a lightweight synchronization scheme, such that queries and updates rarely interfere, and only when necessary. Many challenges arise when combining concurrent operations in this fashion; consistency, freshness, memory efficiency, and operation throughput must all be balanced in a complex multi-way trade-off.

6 Evaluation plan

How are you going to evaluate your research? Which datasets will you use? Are you using any established benchmarks? Have you reached any result so far?

Evaluation is performed in benchmarks using some real-world datasets from the CAIDA packet traces [1], as well as synthetic datasets where we can precisely control parameters such as skew or presence of anomalies.

7 Conclusions and Reflections

What is the current status of your research? What are the main challenges you are facing? Why do you think your research will be successful?

Submitted simultaneously with this, hence the brevity in writing here.

We bring a lot of new insight to this problem that has not been studied previously, and propose a very promising methodology to address the identified challenges.

References

[1] [n. d.]. The CAIDA UCSD Anonymized Internet Traces - 2018. https://www.caida.org/catalog/datasets/passive_dataset

[2] [n. d.]. DataSketches | Apache® DataSketches. https://datasketches.apache.org/

[3] Noga Alon, Yossi Matias, and Mario Szegedy. 1999. The Space Complexity of Approximating the Frequency Moments. *J. Comput. System Sci.* 58, 1 (Feb. 1999), 137–147. doi:10.1006/jcss.1997.1545

[4] Monica Chiosa, Thomas B. Preußer, and Gustavo Alonso. 2021. SKT: A One-Pass Multi-Sketch Data Analytics Accelerator. *Proceedings of the VLDB Endowment* 14, 11 (July 2021), 2369–2382. doi:10.14778/3476249.3476287

[5] Graham Cormode. 2023. Applications of Sketching and Pathways to Impact. In *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS '23)*. Association for Computing Machinery, New York, NY, USA, 5–10. doi:10.1145/3584372.3589937

[6] Graham Cormode and Minos Garofalakis. 2005. Sketching Streams through the Net: Distributed Approximate Query Tracking. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB '05)*. VLDB Endowment, Trondheim, Norway, 13–24.

[7] Graham Cormode and S. Muthukrishnan. 2005. An Improved Data Stream Summary: The Count-Min Sketch and Its Applications. *Journal of Algorithms* 55, 1 (April 2005), 58–75. doi:10.1016/j.jalgor.2003.12.001

[8] Yiannis Nikolakopoulos, Anders Gidenstam, Marina Papatriantafilou, and Philippas Tsigas. 2015. A Consistency Framework for Iteration Operations in Concurrent Data Structures. In *2015 IEEE International Parallel and Distributed Processing Symposium*. 239–248. doi:10.1109/IPDPS.2015.84

[9] Savannah Norem. 2022. Probabilistic Data Structures in Redis. https://redis.io/blog/streaming-analytics-with-probabilistic-data-structures/

[10] Wieger R. Punter, Odysseas Papapetrou, and Minos Garofalakis. 2023. OmniSketch: Efficient Multi-Dimensional High-Velocity Stream Analytics with Arbitrary Predicates. *Proc. VLDB Endow.* 17, 3 (Nov. 2023), 319–331. doi:10.14778/3632093.3632098

[11] Arik Rinberg and Idit Keidar. 2023. Intermediate Value Linearizability: A Quantitative Correctness Criterion. *J. ACM* 70, 2 (April 2023), 17:1–17:21. doi:10.1145/3584699

[12] Charalampos Stylianopoulos, Ivan Walulya, Magnus Almgren, Olaf Landsiedel, and Marina Papatriantafilou. 2020. Delegation Sketch: A Parallel Design with Support for Fast and Accurate Concurrent Operations. In *Proceedings of the Fifteenth European Conference on Computer Systems (EuroSys '20)*. Association for Computing Machinery, New York, NY, USA, 1–16. doi:10.1145/3342195.3387542