CHALMERS
UNIVERSITY OF TECHNOLOGY

Motivations and challenges for stream processing in edge computing

Citation for the original published paper (version of record):
Gulisano, V. (2021). Motivations and challenges for stream processing in edge computing. ICPE 2021 - Companion of the ACM/SPEC International Conference on Performance Engineering: 17-18. http://dx.doi.org/10.1145/3447545.3451899

Motivations and Challenges for Stream Processing in Edge Computing

Vincenzo Gulisano
Chalmers University of Technology, Gothenburg, Sweden
vincenzo.gulisano@chalmers.se

ABSTRACT

The 2030 Agenda for Sustainable Development of the United Nations General Assembly defines 17 development goals to be met for a sustainable future. Goals such as *Industry, Innovation and Infrastructure* and *Sustainable Cities and Communities* depend on digital systems. As a matter of fact, billions of Euros are invested into digital transformation within the European Union, and many researchers are actively working to push state-of-the-art boundaries for techniques/tools able to extract value and insights from the large amounts of raw data sensed in digital systems.

Edge computing aims at supporting such data-to-value transformation. In digital systems that traditionally rely on central data gathering, edge computing proposes to push the analysis towards the devices and data sources, thus leveraging the large cumulative computational power found in modern distributed systems. Some of the ideas promoted in edge computing are not new, though. Continuous and distributed data analysis paradigms such as stream processing have argued about the need for smart distributed analysis for basically 20 years. Starting from this observation, this talk covers a set of standing challenges for smart, distributed, and continuous stream processing in edge computing, with real-world examples and use-cases from smart grids and vehicular networks.

CCS CONCEPTS

• **Information systems** → **Stream management**; **Data streams**; **Online analytical processing engines**;

KEYWORDS

Data streaming; Edge Computing

ACM Reference Format:
Vincenzo Gulisano. 2021. Motivations and Challenges for Stream Processing in Edge Computing. In *Companion of the 2021 ACM/SPEC International Conference on Performance Engineering (ICPE '21 Companion), April 19–23, 2021, Virtual Event, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3447545.3451899

1 TALK OVERVIEW

The talk begins by introducing edge computing [19], discussing both the possibilities it enables as well as the challenges to be addressed, focusing in particular on the stream processing paradigm and its benefits in large distributed systems [3, 20]. The talk continues with a deeper look into stream processing basic concepts, and then focuses on three main challenges, as briefly introduced next.

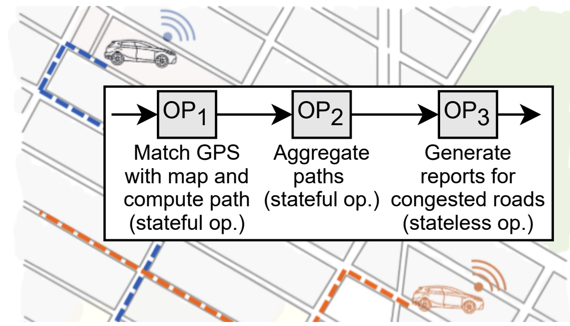

Figure 1: Sample streaming application – traffic monitoring.

Stream Processing in a nutshell. Stream processing is leveraged in large distributed systems [1, 2, 7, 9, 10, 13, 22, 23] to process unbounded *streams* of *tuples*. Stream processing applications are defined as Directed Acyclic Graphs of operators that transform the tuples delivered by a set of data sources and produce new streams of tuples that are eventually delivered to end-users. Operators are either *stateless* and *stateful*. Stateless operators process each input tuple individually and do not maintain a state that evolves according to the tuples being processed. Stateful operators produce results that depend on portions of tuples called *windows* [1]. Figure 1 shows a sample streaming application for traffic monitoring in urban environments. The application ingests GPS position reports and, by locating them on a map, estimates the path being followed by each car. Subsequently, the application aggregates the different paths, creating reports for road segments that are congested.

Challenge 1: There is more to "pushing the analysis to the edge" than meets the eye. What does it mean to *"push the analysis to the edge"* for a streaming application? Naively, one can interpret this as *"pushing some of its operators to the edge"*. This can be beneficial, for instance by having each car locally compute its path concerning the sample application in Figure 1. Besides the obvious limitations imposed by operators that ingest data from multiple sources and cannot be moved arbitrarily close to the latter (e.g., operator OP_2, that needs to aggregate multiple paths and cannot thus run locally within each car accessing only the latter's data), such an approach does not account for alternative ways of leveraging edge devices (and their computational power) [4, 9, 16, 18]. In our example, OP_1 could for instance be replaced by operators OP_{1a}, outputting a tuple carrying a road segment id every time a car enters a new road segment, and OP_{1b}, aggregating into paths such tuples on a per-car basis. Deploying OP_{1a} at each car (rather

ICPE '21 Companion, April 19–23, 2021, Virtual Event, France

ACM ISBN 978-1-4503-8331-8/21/04.
https://doi.org/10.1145/3447545.3451899

than OP_1) could still significantly reduce the data volumes collected from the cars. Moreover, it would allow a utility to run OP_{1b} centrally, updating its semantics (e.g., deciding how many past tuples to aggregate into paths) without extra communication with cars.

Challenge 2: Hardware-, data- and system-awareness. Large distributed systems such as smart grids or vehicular networks are composed of heterogeneous devices and sensors [5, 6, 14, 15, 17, 21], ranging from the small embedded ones found in smart meters to GPU-based platforms for AI-based self-driving cars [9]. Such heterogeneity must be taken into account for streaming applications to scale while sharing resources with the existing ecosystem of applications running in each device. While small amounts of memory suffice for a smart meter to process its consumption readings, the on-board computer of a car can potentially access tens of gigabytes of data per hour [9], while running critical applications such as lane departure warning systems. Continuing our example, running OP_1 (or OP_{1a}) within each car could be conditional to the ability to limit the maximum amount of resources such operator has access to, for instance with a dedicated scheduler [17] or by scaling down the analysis and providing approximate answers [4, 9, 11, 12].

Challenge 3: Model-driven adaptiveness. Adaptive reconfigurations for streaming applications have been proposed since the introduction of the first pioneer streaming frameworks, with techniques such as operator placement, thread scheduling, load balancing, load shedding, and elasticity. These techniques aim at controlling the behavior of performance metrics such as throughput, latency, memory, and CPU consumption. The rapidly evolving nature of edge devices and their data demands for proactive analysis of how such metrics can behave in the short, mid, and long term. Towards this end, proper modeling of streaming operators' behavior can allow for very precise estimation of such metrics with minimal monitoring costs, in accordance with the limited computational resources of devices [18] and the impact monitoring has on such resources [8].

About the speaker

Vincenzo Gulisano (Ph.D.) is an Associate Professor at the Computer Science and Engineering department of Chalmers University of Technology. Dr. Vincenzo Gulisano holds a Ph.D. in Computer Science from the Polytechnic University of Madrid, Spain. His research focuses mainly on stream processing and its use in cyber-physical systems such as smart grids and vehicular networks, covering aspects such as (1) distributed, parallel, and elastic processing capabilities, (2) streaming-aware data structures, (3) edge computing, taking into account security and privacy aspects too, (4) scheduling and use of resources, (5) stream processing provenance, and (6) porting of ML analysis from batch to stream processing.

Acknowledgements

Part of the research discussed in this talk has been financed by VINNOVA, the Swedish Government Agency for Innovation Systems, project "Automotive Stream Processing and Distributed Analytics (AutoSPADA)" in the funding program FFI: Strategic Vehicle Research and Innovation (DNR 2019-05884); the Swedish Foundation for Strategic Research, project "Future factories in the cloud (FiC)" (grant GMT14-0032), and the Swedish Research Council (Vetenskapsrådet), project "HARE: Self-deploying and Adaptive Data Streaming Analytics in Fog Architectures" (grant 2016-03800).

REFERENCES

[1] D. J. Abadi, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik. Aurora: A new model and architecture for data stream management. In *VLDB Journal*, volume 12, pages 120–139, aug 2003.

[2] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 1–16, 2002.

[3] P. Bonnet, J. Gehrke, and P. Seshadri. Towards sensor database systems. In *International Conference on mobile Data management*, pages 3–14. Springer, 2001.

[4] R. Duvignau, V. Gulisano, M. Papatriantafilou, and V. Savic. Streaming piecewise linear approximation for efficient data management in edge computing. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 593–596, 2019.

[5] P. R. Geethakumari, V. Gulisano, B. J. Svensson, P. Trancoso, and I. Sourdis. Single window stream aggregation using reconfigurable hardware. In *2017 International Conference on Field Programmable Technology (ICFPT)*, pages 112–119. IEEE, 2017.

[6] P. R. Geethakumari, V. Gulisano, P. Trancoso, and I. Sourdis. Time-swad: A dataflow engine for time-based single window stream aggregation. In *2019 International Conference on Field-Programmable Technology (ICFPT)*, pages 72–80. IEEE, 2019.

[7] V. Gulisano, M. Almgren, and M. Papatriantafilou. When smart cities meet big data. *Smart Cities*, 1(98):40, 2014.

[8] V. Gulisano, A. V. Papadopoulos, Y. Nikolakopoulos, M. Papatriantafilou, and P. Tsigas. Performance modeling of stream joins. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems*, pages 191–202, 2017.

[9] B. Havers, R. Duvignau, H. Najdataei, V. Gulisano, A. C. Koppisetty, and M. Papatriantafilou. Driven: a framework for efficient data retrieval and clustering in vehicular networks. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1850–1861. IEEE, 2019.

[10] J. H. Hwang, U. Çetintemel, and S. Zdonik. Fast and reliable stream processing over wide area networks. In *Proceedings - International Conference on Data Engineering*, pages 604–613, 2007.

[11] A. Keramatian, V. Gulisano, M. Papatriantafilou, and P. Tsigas. Parma-cc: Parallel multiphase approximate cluster combining. In *Proceedings of the 21st International Conference on Distributed Computing and Networking*, pages 1–10, 2020.

[12] A. Keramatian, V. Gulisano, M. Papatriantafilou, and P. Tsigas. Mad-c: Multi-stage approximate distributed cluster-combining for obstacle detection and localization. *Journal of Parallel and Distributed Computing*, 147:248–267, 2021.

[13] C. Mutschler and M. Philippsen. Distributed low-latency out-of-order event processing for high data rate sensor streams. In *Proceedings - IEEE 27th International Parallel and Distributed Processing Symposium, IPDPS 2013*, pages 1133–1144, 2013.

[14] H. Najdataei, Y. Nikolakopoulos, V. Gulisano, and M. Papatriantafilou. Continuous and parallel lidar point-cloud clustering. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 671–684. IEEE, 2018.

[15] Y. Nikolakopoulos, M. Papatriantafilou, P. Brauer, M. Lundqvist, V. Gulisano, and P. Tsigas. Highly concurrent stream synchronization in many-core embedded systems. In *Proceedings of the Third ACM International Workshop on Many-core Embedded Systems*, pages 2–9, 2016.

[16] D. Palyvos-Giannas, V. Gulisano, and M. Papatriantafilou. GeneaLog: Fine-grained data streaming provenance in cyber-physical systems. *Parallel Computing*, 89:102552, nov 2019.

[17] D. Palyvos-Giannas, V. Gulisano, and M. Papatriantafilou. Haren: A framework for ad-hoc thread scheduling policies for data streaming applications. In *Proceedings of the 13th ACM International Conference on Distributed and Event-based Systems*, pages 19–30, 2019.

[18] D. Palyvos-Giannas, B. Havers, M. Papatriantafilou, and V. Gulisano. Ananke: a streaming framework for live forward provenance. *Proceedings of the VLDB Endowment*, 14(3):391–403, 2020.

[19] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

[20] M. Stonebraker, U. Çetintemel, and S. Zdonik. The 8 requirements of real-time stream processing. *ACM Sigmod Record*, 34(4):42–47, 2005.

[21] J. van Rooij, V. Gulisano, and M. Papatriantafilou. Locovolt: Distributed detection of broken meters in smart grids through stream processing. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, pages 171–182, 2018.

[22] I. Walulya, D. Palyvos-Giannas, Y. Nikolakopoulos, V. Gulisano, M. Papatriantafilou, and P. Tsigas. Viper: A module for communication-layer determinism and scaling in low-latency stream processing. *Future Generation Computer Systems*, 88:297–308, 2018.

[23] N. Zacheilas, V. Kalogeraki, Y. Nikolakopoulos, V. Gulisano, M. Papatriantafilou, and P. Tsigas. Maximizing determinism in stream processing under latency constraints. In *DEBS 2017 - Proceedings of the 11th ACM International Conference on Distributed Event-Based Systems*, pages 112–123, New York, New York, USA, jun 2017. Association for Computing Machinery, Inc.