

# Towards Efficient and Secure UDF Execution with BabelfishLib (Lightning Talk)

Philipp M. Grulich<sup>1,\*</sup>, Steffen Zeuch<sup>1</sup> and Volker Markl<sup>1,2</sup>

<sup>1</sup>Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin

<sup>2</sup>DFKI GmbH, Alt-Moabit 91c, 10559 Berlin

## Abstract

Today, data scientists, web developers, and application developers build complex data processing pipelines by combining different tools and programming languages. To this end, most data processing systems support user-defined functions (UDFs) in common languages like Java, Python, or JavaScript. These UDFs enable users to express arbitrary business logic in their preferred programming language, leverage 3rd-party libraries, and increase the modularity and testability of their data processing pipelines. Although UDFs provide a large degree of freedom, their flexibility comes with a high-performance cost compared to traditional relational queries. As a result, most experts recommend avoiding UDFs whenever possible.

To cope with these inefficiencies, research has suggested several strategies. These include translating UDFs to semantically equivalent SQL statements, extending optimizers to the unique properties of UDFs, and devising efficient execution strategies that mitigate the bottlenecks of UDFs. These approaches, while delivering performance improvements, necessitate substantial engineering efforts and amplify system complexity, which hinders their widespread adoption. To improve this situation, we propose in this talk BabelfishLib, which provides our Babelfish Engine [1] as an extensible component for the efficient and secure execution of UDFs. In an environment where virtually every data management system requires UDF support, BabelfishLib can centralize these efforts and provide a unified UDF runtime that can be used across different systems. In particular, BabelfishLib targets three major design goals. First, it provides efficient execution strategies for UDFs in different programming languages. Second, it ensures that the execution of untrusted UDF code is isolated from the data processing system, guaranteeing system security. Third, it analyzes UDFs and provides information for further query optimizations. As a result, BabelfishLib mitigates the performance overhead of UDFs in state-of-the-art systems while it ensures security and isolation at the same time. Currently, we leverage BabelfishLib to accelerate UDFs in our data processing platform NebulaStream [2, 3].

We believe that BabelfishLib can be a first step towards a unified accelerator for UDFs, which can be integrated across different data processing systems. Furthermore, it provides a playground for further research focusing on specific aspects of the acceleration of UDF. Finally, through this presentation, we intend to spark a discussion across the community to consolidate requirements for efficient UDF execution and combine different efforts in the same direction.

## Keywords

User-Defined Functions, Data Management

## References

- [1] P. M. Grulich, S. Zeuch, V. Markl, Babelfish: Efficient execution of polyglot queries, Proc. VLDB Endow. 15 (2021) 196–210. URL: <https://doi.org/10.14778/3489496.3489501>. doi:10.14778/3489496.3489501.
- [2] S. Zeuch, X. Chatziliadis, A. Chaudhary, D. Giouroukis, P. M. Grulich, D. P. A. Nugroho, A. Ziehn, V. Markl, NebulaStream: Data management for the internet of things, Datenbank-Spektrum 22 (2022) 131–141. doi:10.1007/s13222-022-00415-0.
- [3] S. Zeuch, E. T. Zacharitou, S. Zhang, X. Chatziliadis, A. Chaudhary, B. Del Monte, D. Giouroukis, P. M. Grulich, A. Ziehn, V. Markl, NebulaStream: Complex analytics beyond the cloud, Open Journal of Internet Of Things (OJIOT) 6 (2020) 66–81. URL: [https://www.ronpub.com/ojiot/OJIOT\\_2020v6i1n07\\_Zeuch.html](https://www.ronpub.com/ojiot/OJIOT_2020v6i1n07_Zeuch.html).

Joint Workshops at 49th International Conference on Very Large Data Bases (VLDBW'23) – Second International Workshop on Composable Data Management Systems (CDMS'23), August 28 - September 1, 2023, Vancouver, Canada

✉ grulich@tu-berlin.de (P. M. Grulich); steffen.zeuch@tu-berlin.de (S. Zeuch); volker.markl@tu-berlin.de (V. Markl)

🌐 <https://grulich.me/> (P. M. Grulich)

📄 0000-0001-9497-2895 (P. M. Grulich); 0000-0002-4082-7788

(S. Zeuch); 0009-0009-0964-026X (V. Markl)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)