(Lightning) Towards a Practical HTAP Engine for Polystores

Hiroyuki Yamada, Akihiro Okuno, Toshihiro Suzuki and Jun Nemoto

Scalar, Inc. Tokyo, Japan

Abstract
In this Lightning talk, we introduce a ScalarDB approach for achieving HTAP in polystores.

Keywords
polystores, federated database systems, HTAP

1. Introduction

Polystores have gained much interest in the database community because of an increasing need for combining a variety of special-purpose databases, modernizing system architecture (e.g., employing a microservice architecture [1, 2]), and dealing with siloed databases in enterprises [3, 4].

As one of the approaches to realizing polystores from a transactional perspective, we have been developing and providing a product called ScalarDB [5]. ScalarDB achieves distributed transactions across multiple disparate databases. Specifically, ScalarDB provides a database-agnostic transaction manager on top of its database abstraction; thus, it achieves transactions spanning multiple disparate databases without depending on the transactional capability of underlying databases. Although ScalarDB was started as a universal transaction manager, it has evolved as a universal HTAP engine, running complex analytical workloads on the databases with which ScalarDB interacts while providing transactional capability.

This lightning talk introduces the ScalarDB approach for HTAP in polystores. To provide the HTAP capability, ScalarDB integrates PostgreSQL foreign data wrappers (FDW). ScalarDB with PostgreSQL FDW provides a unified read-committed view of underlying databases, reading records from underlying databases through community-provided data wrappers or ScalarDB Data Wrapper, selecting committed records by checking the states of the records with a predefined view, and executing a given query in PostgreSQL with the committed records.

This talk also discusses how we plan to extend ScalarDB to achieve more scalable analytical processing to realize a true HTAP engine for polystores. One of the prospective approaches is running existing distributed query engines on top of ScalarDB-managed databases. Specifically, since many existing distributed query engines separate compute from storage to make them cloud-native [6, 7] by providing elegant storage abstraction, we could utilize such storage

Poly'23: Polystore systems for heterogeneous data in multiple databases with privacy and security assurances
hiroyuki.yamada@scalar-labs.com (H. Yamada); akihiro.okuno@scalar-labs.com (A. Okuno); toshihiro.suzuki@scalar-labs.com (T. Suzuki); jun.nemoto@scalar-labs.com (J. Nemoto)

abstraction mechanisms to seamlessly integrate such distributed query engines with ScalarDB-managed databases.

References

[1] J. Lewis, M. Fowler, Microservices, https://martinfowler.com/articles/microservices.html, 2014.

[2] R. Laigner, Y. Zhou, M. Salles, Y. Liu, M. Kalinowski, Data Management in Microservices: State of the Practice, Challenges, and Research Directions, PVLDB 14 (2021) 3348–3361.

[3] B. Hagler, Overcoming Data Silos In Your Organization, https://www.forbes.com/sites/forbestechcouncil/2020/05/12/overcoming-data-silos-in-your-organization, 2020.

[4] P. Strong, Enterprise Grid Computing: Grid Computing Holds Great Promise for the Enterprise Data Center, but Many Technical and Operational Hurdles Remain., Queue 3 (2005) 50–59.

[5] H. Yamada, T. Suzuki, Y. Ito, J. Nemoto, ScalarDB: Universal Transaction Manager for Polystores, PVLDB 16 (2023).

[6] The Apache Software Foundation, Apache Spark, https://spark.apache.org/, 2023.

[7] The Trino Software Foundation, Trino, https://trino.io/, 2023.