

The Case for Decoupled Transaction Managers (Lightning Talk)

Hiroyuki Yamada, Toshihiro Suzuki, Vincent Guilpain, Mitsunori Komatsu, Akihiro Okuno and Jun Nemoto

Scalar, Inc. Tokyo, Japan

Abstract

In this talk, we discuss the case for decoupling transaction managers from databases, which achieves a unique benefit in addition to the reusability of transaction managers, such as it naturally realizes transactions across multiple disparate databases, i.e., global transactions. We also introduce a case study with ScalarDB, a decoupled transaction manager that achieves global transactions.

Keywords

transaction management, database architecture, federated database systems, polystores

A transaction manager is the central component of OLTP databases. It is commonly tightly coupled with other database components, such as a buffer manager and a log manager, because of transaction performance efficiency.

In this talk, we discuss the case for decoupling transaction managers from databases. Although decoupling transaction managers has several downsides, such as making it challenging to optimize transaction performance, it achieves a unique benefit in addition to the reusability of transaction managers, such as it naturally achieves transactions across multiple disparate databases (i.e., global transactions), which is very difficult without decoupling transaction managers.

Achieving global transactions is attractive in many cases: First, *one size does not fit all* is becoming common sense in data management systems. Major cloud vendors offer several purpose-built database products to meet various users' needs [1]. Unsurprisingly, there are many cases where an application uses multiple databases to provide its services. Second, a microservice architecture accelerates the trend of managing multiple (potentially disparate) databases [2]. Each microservice of a single application is encouraged to use an isolated database, which is selected based on the service's use cases and the

developers' experiences for better maintainability and productivity. This architectural style is likely to make an application have different kinds of databases or multiple instances of the same database. Third, managing multiple disparate databases is not uncommon in enterprise systems as well [3]. An enterprise comprises several organizations, departments, and business units to support agile business operations. This leads to siloed information systems; different organizations manage different applications at disparate locations, and the applications use different databases.

In this talk, we also introduce a case study with ScalarDB [4], a decoupled transaction manager that achieves global transactions. 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.

References

- [1] A. W. Services, *Aws cloud databases*, <https://aws.amazon.com/products/databases/>, 2023.
- [2] J. Lewis, M. Fowler, *Microservices*, <https://martinfowler.com/articles/microservices.html>, 2014.
- [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] H. Yamada, T. Suzuki, Y. Ito, J. Nemoto, *ScalarDB: Universal Transaction Manager for Polystores*, *PVLDB* 16 (2023) 3768–3780.

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

✉ hiroyuki.yamada@scalar-labs.com (H. Yamada);
toshihiro.suzuki@scalar-labs.com (T. Suzuki);
vincent.guilpain@scalar-labs.com (V. Guilpain);
mitsunori.komatsu@scalar-labs.com (M. Komatsu);
akihiro.okuno@scalar-labs.com (A. Okuno);
jun.nemoto@scalar-labs.com (J. Nemoto)

ORCID 0000-0002-7137-0899 (H. Yamada)

© 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)