Technical Perspective for Skeena: Efficient and Consistent Cross-Engine Transactions

[Research Highlight]

Carsten Binnig
Technical University of Darmstadt & DFKI
carsten.binnig@tu-darmstadt.de

1. MULTI-ENGINE DATABASE SYSTEMS

The paper proposes a solution to the problem of inadequate support for transactions in multi-engine database systems. Multi-engine database systems are databases that integrate new (fast) memory-optimized storage engines with (slow) traditional engines, allowing the application to use tables in both engines. Multi-engine database systems are in particular interesting for traditional database systems that are extended over time. By being able to store tables in slow and fast storage engines and executing transactions cross engines allows to reduce overall cost since less performance critical tables can be placed in slow (and thus cheaper) storage.

As a main contribution, the paper introduces Skeena - an approach to enable efficient and consistent cross-engine transactions in multi-versioned fast-slow systems. Skeena consists of a cross-engine snapshot registry (CSR) and an extended pipelined commit protocol for atomicity and durability without traditional two-phase commit (2PC). CSR maintains mappings between commit timestamps (snapshots) in one engine and those in another, allowing for efficient snapshot selection. The extended pipelined commit protocol ensures atomicity and durability while imposing low overhead, especially on the faster engine.

2. CHALLENGES AND KEY IDEAS

The paper discusses the challenges associated with existing cross-engine transaction support, including correctness, performance, and programmability. Existing solutions are inadequate as they do not ensure correct cross-engine execution, ignore the performance imbalance between the two engines and require non-trivial application changes that limit functionality.

The authors thus make three key observations to guide Skeena's design. First, inconsistent snapshots can be avoided by carefully selecting a snapshot in each engine. Second, commit ordering can be used to ensure serializability, and concurrency control protocols exhibit this property. Finally, in a multi-engine DBMS, vendors can still open-up the internals of the individual engines to allow for more aggressive optimizations when running cross-engines transactions.

Skeena addresses these challenges by allowing for correct and efficient snapshot selection, ensuring atomicity and durability without traditional 2PC, and imposing low overhead, especially on the faster memory-optimized engine. Skeena can be easily plugged into an existing system and leverages the fact that engines can communicate via fast shared memory. Overall, Skeena offers an elegant solution to the challenges associated with cross-engine transactions in multi-versioned fast-slow systems.

3. WHY READING THE PAPER?

I enjoyed reading the paper very much. It combines several aspects that make it an interesting read.

First, Skeena opens up an alternate route to other recent designs such as LeanStore [2] or Umbra [3]. LeanStore or Umbra argue that due to the high main memory cost, we should rather focus on building fast on-SSD storage engines and database systems instead of focusing on pure in-memory designs. With a cross-engine design, we can achieve similar goals as LeanStore or Umbra since not all data needs to be placed in expensive main memory.

Second, Skeena can be easily adopted by real systems, as shown by the paper with MySQL. Moreober, the evaluation on a 40-core server shows that Skeena incurs negligible overhead and maintains the benefits of memory-optimized engines which was one of the main challenges the paper aimed to address.

Finally, Skeena also builds nicely on previous results [1] which have proposed similar techniques to optimize transaction execution in distributed databases across nodes acting as individual engines. Skeena uses the ideas of this prior work and optimizes them for the use in a single machine.

4. REFERENCES

[1] C. Binnig, S. Hildenbrand, F. Färber, D. Kossmann, J. Lee, and N. May. Distributed snapshot isolation: global transactions pay globally, local transactions pay locally. *VLDB J.*, 23(6):987–1011, 2014.

[2] V. Leis, M. Haubenschild, A. Kemper, and T. Neumann. Leanstore: In-memory data management beyond main memory. In *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*, pages 185–196. IEEE Computer Society, 2018.

[3] T. Neumann and M. J. Freitag. Umbra: A disk-based system with in-memory performance. In *10th Conference on Innovative Data Systems Research, CIDR 2020, Amsterdam, The Netherlands, January 12-15, 2020, Online Proceedings.* www.cidrdb.org, 2020.