Evaluating Deterministic Database Sequencers

Oto Mraz, Kyriakos Psarakis, George Christodoulou, and Asterios Katsifodimos
TU Delft
{o.m.mraz,k.psarakis,g.c.christodoulou,a.katsifodimos}@tudelft.nl

With the increasing abundance of financial, retail or logistics companies operating on a cross-continental scale, the design of efficient distributed deterministic databases becomes increasingly critical. In such setups, the throughput, latency and consistency of database transactions are paramount. Speed and consistency fall under the responsibility of the *sequencer*, the component in charge of ordering incoming transactions. If the whole database resides in a single datacenter, ordering transactions is trivial and can be executed with minimal delay. However, a single datacenter setup scales poorly and provides no fault tolerance in the event of a datacenter failure. Thus, database architects advocate for geo-partitioned databases, where data is stored close to its respective clients.

Despite the clear scalability and resilience benefits, geo-partitioned database sequencers also face challenges due to the high communication overheads, data transfer costs, and the risk of transaction aborts, especially when transactions touch data from multiple regions [3]. In practice, an efficient sequencer must also cope with diverse and changing conditions, e.g., unstable/slow networks, frequent failures, fluctuating transaction load. The state-of-the-art sequencers address these challenge in two fundamental ways: 1) Reduce the amount of cross-region communication needed to agree on the transaction order. 2) Shorten the critical path, that requires locking, to reduce the likelihood of a conflict in the first place [2].

Our preliminary research indicates that no single sequencing architecture optimally handles all system conditions and workloads. It has been shown that a sequencer's performance varies based on factors including the ratio of multi-region transactions, conflict rates, and hardware limitations (CPU, memory, network) [1]. Moreover, using synchronous replication as a fault-tolerance strategy improves the resilience and failure recovery speed but adds additional complexity and overhead to the sequencer.

Our current work focuses on developing a unifying taxonomy of deterministic database sequencers and evaluating their performance under diverse working conditions. We assess not only the sequencers' trade-offs in terms of throughput and latency, but also gauge their aborts, network demands, cost, and fault tolerance. We aim to guide the design of adaptable, pluggable sequencer architectures that adapt dynamically to the needs of specific workloads.

References

[1] R. Harding et al. "An evaluation of distributed concurrency control". In: *Proc. VLDB Endowment* 10.5 (2017), pp. 553–564.

[2] C. D. Nguyen, J. K. Miller, and D. J. Abadi. "Detock: High performance multi-region transactions at scale". In: *Proc. ACM on Management of Data* 1.2 (2023), pp. 1–27.

[3] K. Ren, D. Li, and D. J. Abadi. "SLOG: Serializable, low-latency, geo-replicated transactions". In: *Proc. VLDB Endowment* 12.11 (2019).