Eventual Durability of ACID Transactions in Distributed Database Systems

Joseph Boulis, Sujaya Maiyya, Ken Salem

David R. Cheriton School of Computer Science
University of Waterloo
{jboulis,smaiyya,ken.salem}@uwaterloo.ca

1 Abstract

Distributed transactions are critical for scaling across sharded systems but often suffer from high latency. This latency is primarily attributed to the overhead involved in guaranteeing durability, particularly when leveraging replicated logs [1, 3, 5]. To address this, we aim to provide flexibility in managing performance/durability tradeoffs for distributed transactions. The main challenge arises from the traditional transactional model, which couples durability with commitment. This model defines a single point at which a transaction becomes visible and durable, requiring all transaction guarantees to wait for the slowest component—typically durability.

In this work, we explore how to implement the Eventual Durability (ED) transaction model [2] in a distributed setting. The ED model decouples commit from durability, allowing transactions to commit first, and become durable later. In essence, committed but non-durable transactions can be visible to applications, enabling operations to continue while ensuring eventual durability in the background. This approach enables informed decisions about sacrificing immediate durability for lower latencies while understanding risks of system failures.

Adapting the ED model to distributed settings presents unique challenges, particularly as partial transaction failures across shards complicate durability and isolation guarantees. In centralized systems, if transaction T2 depends on T1, T2 will fail if T1 does. In distributed settings, however, T2 may commit on one shard while T1 fails on another, leading to inconsistencies. To address these challenges, we introduce an eventually durable Percolator protocol [4], enabling applications to reduce transaction latencies while preserving guarantees. We advocate for this model as the foundational contract between transactional data systems and applications, outlining requirements for implementing eventually durable transactions and their effective exposure to applications.

2 Preliminary Results

We conducted a preliminary experiment on AWS with a setup involving two shards and writing to two keys. Each trial included an average of 10,000 transactions. The results, as shown in Table 1, demonstrate that the ED model outperforms the traditional transaction model by achieving a significantly lower commit latency. This improvement is due to the absence of commit time waiting in the ED model, resulting in approximately a 67% reduction in latency compared to the traditional model.

Transaction Model	Average Latency (ms)
ED (Experimental Data)	4.84
Traditional Transaction Model	14.5

Table 1: Commit Latency Comparison between ED and Traditional Transaction Model

References

[1] Dongxu Huang et al. "TiDB: a Raft-based HTAP database". In: *Proceedings of the VLDB Endowment* 13.12 (2020), pp. 3072–3084.

[2] Tejasvi Kashi. "Eventual Durability of ACID Transactions in Database Systems". MA thesis. University of Waterloo, 2023.

[3] Wei Lin et al. "PacificA: Replication in log-based distributed storage systems". In: (2008).

[4] Daniel Peng and Frank Dabek. "Large-scale incremental processing using distributed transactions and notifications". In: *9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10)*. 2010.

[5] Rebecca Taft et al. "Cockroachdb: The resilient geo-distributed sql database". In: *Proceedings of the 2020 ACM SIGMOD international conference on management of data*. 2020, pp. 1493–1509.