Fragola: Low-Latency Transactions in Distributed Data Stores

Yonatan Gottesman* Aran Bergman† Edward Bortnikov*
Eshcar Hillel* Idit Keidar* † Ohad Shacham*
* Yahoo Research † Technion

ACM Reference Format:
Yonatan Gottesman[1] Aran Bergman[2] Edward Bortnikov[1] Eshcar Hillel[1] Idit Keidar[1] [2] Ohad Shacham[1] [1] Yahoo Research [2] Technion . 2017. Fragola: Low-Latency Transactions in Distributed Data Stores. In *Proceedings of SoCC '17, Santa Clara, CA, USA, September 24–27, 2017,* 1 pages.
https://doi.org/10.1145/3127479.3132686

As transaction processing services begin to be used in new application domains, low transaction latency becomes an important consideration. Motivated by such use cases we developed Fragola, a highly scalable low-latency *and* high-throughput transaction processing engine for Apache HBase. Similarly to other modern transaction managers, Fragola provides a variant of *generalized snapshot isolation (SI)*, which scales better than traditional serializability implementations.

Fragola is based on Apache Omid [1], but dissipates the principal bottleneck present therein. Its advantage is maximized for short transactions, which are prevalent in latency-sensitive applications. Fragola processes such transactions in a handful of milliseconds. The new protocol also doubles the system throughput.

As a separate contribution, we introduce a novel *fast path* algorithm for short single-key transactions that eliminates the begin/commit overhead entirely, and executes short transactions almost as fast as native HBase operations. This entails minor extensions to the underlying data store. The fast path is orthogonal to other protocol aspects, and can be supported in other transaction processing services.

We have implemented Fragola based on the open source Omid code[1], and extended the HBase code to enable fast path transactions[2]. Our experiments on mid-range hardware show substantial performance improvements. For example, Figure 1 shows the performance of single-operation transactions using Omid and two variants of Fragola: Vanilla Fragola, which does not include the fast path support, and FP Fragola, which includes the fast path. We can see that under low load (left side of the graph), even without the fast path, Fragola transactions are 4x to 5x faster than Omid's, and the fast path further reduces the latency of short transactions by another 55% on average. As system load increases, Omid's latency surges (at ~150K tps in our tests), whereas Fragola's remains stable until we generate a higher load (~250K tps), and increases four-fold at 500K tps. Overall fast path transactions incur no scalability bottlenecks, and continue to execute at the low latency of native HBase operations regardless of system load (thanks to HBase's near perfect horizontal scalability). This comes at the cost of a minor (less than 15%) negative impact on longer transactions. Additionally, Fragola has negligible impact on transaction abort rates.

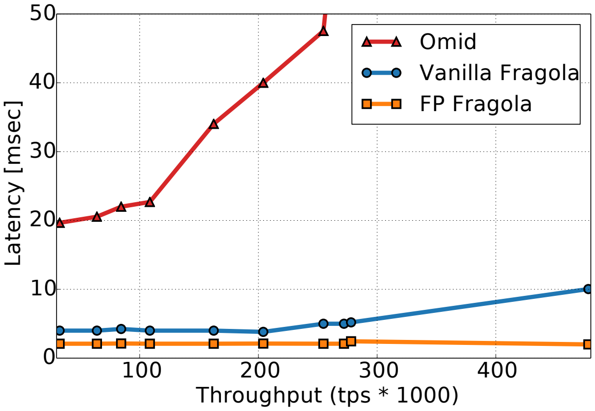

Figure 1: Throughput vs. latency, transaction size = 1. Vanilla Fragola does not include the fast path.

The Fragola code is publicly available[3], and we are working to contribute it back to Omid. We believe our improvements will become instrumental for modern cloud-based, large-scale, latency-sensitive services, e.g., the Phoenix OLTP system, which is designed to harness up to ten thousand nodes.

REFERENCES

[1] O. Shacham, F. Perez-Sorrosal, E. Bortnikov, E. Hillel, I. Keidar, I. Kelly, M. Morel, and S. Paranjpye. Omid, reloaded: Scalable and highly-available transaction processing. In *15th USENIX Conference on File and Storage Technologies (FAST)*, 2017.

[1] https://omid.incubator.apache.org
[2] https://github.com/yonigottesman/hbase_local_transactions/tree/0.98-add-rmw
[3] https://github.com/yonigottesman/incubator-omid/tree/localTransactions

SoCC '17, September 24–27, 2017, Santa Clara, CA, USA
© 2017 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5028-0/17/09.
https://doi.org/10.1145/3127479.3132686