**Transtorm: a benchmark suite for transactional key-value storage systems**

Naveen Kr. Sharma®, Brandon Holt®, Irene Zhang®, Dan R. K. Ports®, Marcos K. Aguilera++

®University of Washington  ++VMware Research Group

(Naveen, Brandon, and Irene are students. The poster will be presented by a combination of them)

Key-value storage systems are popular due to their simplicity, performance, and scalability. Such systems are gradually evolving and some recent ones now support transactions, which are a convenient means to coordinate access to data shared among many clients. Examples of such systems include Hyperdix Warp, CockroachDB, Walter, and TAPIR. Because these systems are so new, there are not many ways to assess how well they implement transactions. To address this shortcoming, we propose Transtorm, a benchmark suite for such transactional key-value storage systems (TKVS).

TKVS’s are today generally evaluated using ad-hoc workloads, making it hard to understand and compare the performance characteristics of different systems. To our knowledge, the only existing benchmark designed for transactional key-value storage systems is [YCSB+T]. YCSB+T extends the popular non-transactional YCSB benchmark by bundling a number of the benchmark’s operations into transactions. This extension is simple and intended to preserve the workload characteristics of the original YCSB benchmark, but as a result it does not directly gauge how well a TKVS addresses the various challenges of implementing transactions. At the same time, while there are many transaction processing benchmarks for traditional databases (e.g., TPC-C), they do not target the scale-out workloads for which TKVS’s are designed. Our goal is to design a benchmark to specifically exercise and challenge the TKVS’s transactional mechanisms.

Benchmarks can be of two main flavors. Macro benchmarks are centered around specific applications, while micro benchmarks are centered around specific mechanisms or features. We decided to propose a micro benchmark because TKVS’s are so new that their target applications are still in flux.

To define the Transtorm suite, we identify three main factors that encumber the execution of transactions in a TKVS: overlapping operations, spread, and duration. Overlapping operations refer to operations that access the same data in incompatible ways across transactions; they create a challenge due to the need to detect them and mitigate their effects (e.g., by aborting transactions). Spread is the number of different objects that a transaction accesses; spread correlates to how distributed the transaction is, and the greater the distribution, the higher the execution overhead. And the duration of a transaction creates a challenge because transactional mechanisms (locks, optimistic checks, etc) have overheads that are proportional to the time of their employment.

For each of these three challenging factors, we define a benchmark to measure the TKVS’s ability to cope with that factor. Each benchmark has a parameter that represents the intensity of the challenge. We evaluate these results in the context of a fourth baseline benchmark that assesses the underlying scalability and performance of the TKVS in the absence of these challenging factors.

The Transtorm benchmarks make minimal assumptions about the underlying TKVS, only that there are operations to start a transaction, to read and write key-value pairs, and to try to commit transactions. As a result, the suite is widely applicable. We have specified the benchmark workloads independently of programming language, and are also working on an open source reference implementation with bindings for several popular languages and TKVS’s.

The main purpose of our poster is to request feedback from participants on whether the Transtorm suite adequately covers the performance concerns TKVS users and, if not, how we can extend the suite. This is still work in progress.