**Abstract**

Chevrotain is a key value store that achieves eventual consistency through the use of a conflict-free replicated data types (CRDTs). This paper describes the designs, evaluates performances, and compares three different approaches to the implementation of Chevrotain. One of the approaches is based on a state-based CRDT model (CvRDT), while the other two approaches are based on an operation-based CRDT model (CmRDT), either with or without limited synchronization. Latency of all implementations is measured as a function of throughput, and consistency of all implementations is compared to a replicated key-value store that only makes a minimal effort to maintain consistency. Scalability of all implementations is also studied. Results shows that the unsynchronized CmRDT implementation of Chevrotain demonstrates best performance. In particular, this implementation easily scales to five geographically distant replicas, achieves sub-second latency under a throughput of 100ops/second while maintaining perfect eventual consistency. A proof-of-concept application to distributed web crawling is presented.

**Introduction**

In distributed systems there is always the tension between maintaining consistency and demonstrating high performance. On one end of the spectrum there is strong consistency which implies that the distributed system behaves like a single machine that serializes all operations. Strong consistency maintains perfect data consistency at all times, at the expense of high latency due to the need for synchronization and constant communication between nodes. On the other end of the spectrum is eventual consistency which allows for the states of the distributed replicas to temporarily diverge and meet later in time. Eventual consistency results in a lower frequency of communication which leads to better performance; however, eventual consistency could lead to data inconsistencies and data loss [1].

Some of the recent research focused on trying to strike a balance between strong consistency and high performance. This has led to emergence of mixed consistency semantics, such as RedBlue consistency, consistency rationing, PSI and Horus [1, 2, 3, 4]. In such semantics, strong consistency is enforced on the operations that depend on data to be immediately consistent, while weak consistency is used with operations for which a high degree of consistency is not necessary. An alternate approach is for programmers to design the distributed system in a way that doesn’t require strong consistency at all. This is the approach that is used in concept of conflict-free replicated data types (CRDT) [5], which is the focus of this paper.

In CRDTs, the distributed system is designed such that either the states of replicas could be merged in a conflict-free way at any point in time (state-based RDT or CvRDT) or that any updates to the states of the replicas are only done in a conflict-free way (operation-based RDT or CmRDT) [5]. This paper presents the design, implementation and performance evaluation of Chevrotain, a CRDT-based key-value store. Three different approaches to implementation of Chevrotain are considered. One of the approaches is based on a CvRDT mode, while the other two approaches are based on the CmRDT model, either with or without limited synchronization.

Key-value stores present a very flexible data storage model that has widespread applications, which allows Chevrotain to be appropriate tool in a variety of applications. We present one such application, a distributed web crawler, in section 5.

This paper begins by briefly describing the mathematical foundations behind CRDTs in section 2. The designs of all proposed implementations of Chevrotain are then detailed in section 3. Section 4 presents the performance evaluation results, while sections 5-8 briefly survey applications, related work, future work on this project and present the conclusion.

**Background: CvRDT**