Efficient (re)naming in Conflict-free Replicated Data types (CRDTs)

Matthieu Nicolas, Gérald Oster and Olivier Perrin
Inria, F-54600, Université de Lorraine, LORIA, F-54506, CNRS, F-54506
Contact Author: matthieu.nicolas@inria.fr

Abstract.

In order to design large scale distributed systems, the literature and companies increasingly adopt the optimistic replication model known as eventual consistency to replicate data among nodes. This consistency model allows replicas to temporarily diverge to be able to ensure high availability. Each node owning a copy of the data can edit it without any kind of coordination with other nodes, before propagating the changes to others. A conflict resolution mechanism is thus required to handle updates generated in parallel by different replicas.

An approach which gains in popularity during recent years proposes to define Conflict-free Replicated Data types (CRDTs). These data structures behave as traditional ones, like the *Set* or the *Sequence* data structures, but are designed for a distributed usage. Their specification ensures that concurrent changes are resolved deterministically and that replicas converge eventually after observing all updates.

To achieve convergence, CRDTs proposed in the literature mostly rely on identifiers to reference updated elements. To be globally unique, element identifiers often include the identifier of the node which generates them. But, since node identifiers grow as new nodes join the system, element identifiers have to grow proportionally. Furthermore, element identifiers have to comply to additional constraints according to the CRDT, which may result in the acceleration of their growth.

Hence, since the size of element identifiers is not bounded, the size of metadata attached to each element increases over time. It thus exceeds more and more the size of data itself. This impedes the usage of CRDTs since nodes have to broadcast and store metadata, causing the application's performances to decrease over time.

The goal of this PhD is to address this issue by 1. proposing more efficient specifications of identifiers according to their set of constraints, 2. proposing mechanisms to rename identifiers to reduce their size.

References