Making Operation-based CRDTs Operation-based

[Work in progress report]

Carlos Baquero
HASLab, INESC TEC &
Universidade do Minho
Braga, Portugal
cbm@di.uminho.pt

Paulo Sérgio Almeida
HASLab, INESC TEC &
Universidade do Minho
Braga, Portugal
psa@di.uminho.pt

Ali Shoker
HASLab, INESC TEC &
Universidade do Minho
Braga, Portugal
shokerali@di.uminho.pt

ABSTRACT

Conflict-free Replicated Datatypes can simplify the design of predictable eventual consistency. They can be classified into state-based or operation-based. Operation-based designs have the potential for allowing compact solutions in both the sent message and the object state size, but current approaches are still far from this objective. Here we explore the design space for operation-based solutions, and we leverage the interaction with the middleware by offering a technique that delivers very compact solutions, while only broadcasting operation names and arguments.

Keywords

Eventual Consistency, Operation-based CRDTs.

1. INTRODUCTION

In distributed databases [4], data replication can improve system performance and fault tolerance, but also impact the exposed level of data consistency. Offering the users the impression of an always-available single consistent copy is not easy in the presence of partitions among the replicas [5]. As partitions, communication failures and topology changes are deemed to occur in all but the smallest systems, and since losing availability is normally not an option, developers have successfully explored relaxed consistency models [3], such as eventual consistency [10, 1].

In eventually consistent systems, data replicas are allowed to diverge; however, this divergence can be tracked so that eventually replicas can be reconciled into a common consistent state. In particular, causal consistency makes sure that each replica has access to all the operations that can influence its state. It is also proven, in [7], that no consistency stronger than causal consistency can be provided in an always-available system that eventually converges.

Crafting, by hand, correct *merge* functions that can reconcile divergent replicas is costly and error prone, and errors can compromise eventual consistency. Merge functions depend on the particular semantics of the concrete datatype the replica is storing. For instance, in a replicated counter that is subject to *increment* operations, the objective of the *merge* would be to account for all distinct *increment* operations known to the replicas being merged. Conflict-free replicated datatypes (CRDTs) [8, 9] offer a model for designing correct replicated datatypes that are always-available and are guaranteed to eventually converge once all operations are known to all replicas.

CRDTs have two complementary designs: (a) Operation-based CRDTs ship each received operation to all replicas, typically over reliable causal broadcast to ensure causal consistency. Replicas converge as long as all concurrent operations are allowed to be received in any order. (b) State-based CRDTs ship full state payloads, resulting from applying operations to a local replica state, and have a commutative, associative and idempotent merge function that deterministically reconciles any two replica states. In mathematical terms, state-based CRDTs define a least upper bound, over a join-semilattice.

There is a trade-off between the above two approaches. Operation-based CRDTs can allow for simpler implementations and a simpler replica state, while requiring more guaranties from the message dissemination layer, namely, reliable causal broadcast. In contrast, state-based CRDTs require more complex states, i.e., storing more meta-data; however, they support ad-hoc dissemination of states, and can handle duplicate and out-of-order delivery of state payloads once merged at the destination replicas, without breaking causal consistency.

However, the current definition of operation-based CRDTs is very relaxed and allows for implementations that send extra information beyond to what is needed to identify an operation, e.g., sending sets of unique element identifiers when propagating a remove operation in an observed-removed set. This, makes it confusing to distinguish the difference between the two models, and imposes a notable source of inefficiency induced by this additional information.

In this work we will focus on improving the current model of operation-based CRDTs by leveraging the causal metadata already present in most reliable causal delivery broadcast protocols [2, 6]. The resulting model allows the exchange of small messages (only operation name and arguments) and a very compact state at the replicas. The work includes the following contributions:

- The definition of a more strict version of operation-based CRDTs, that uses small messages that encode an operation name and possible arguments. Hereby denoted as *pure* operation-based CRDTs.

PaPEC'14 April 13-16 2014, Amsterdam, Netherlands

- Identifying which datatypes are possible, and which are not, in the *pure* model, over off-the-shelf reliable broadcast implementations.
- Defining an extended API for reliable causal broadcast that leverages existing metadata and makes it available to the CRDT developer. Denoted as *tagged* reliable causal broadcast.
- Defining simple pure-operation based models over the tagged reliable broadcast support, that make it possible do define new datatypes deemed to be impossible over standard middleware. These simple models allow a clear description of the concurrency semantics of each datatype.
- Introducing an efficient compacting technique that allows for implementations of non-trivial datatypes with a very compact replica state.
- Implementing compact pure operation based CRDTs for sets with *add-wins* concurrent semantics and *Dynamo* style *multi-value* registers.

2. AKNOWLEDGMENTS

We would like to thank Marek Zawirski, Ricardo Gonçalves and anonymous reviewers for comments that helped improve this work. Project Norte-01-0124-FEDER-000058 is co-financed by the North Portugal Regional Operational Programme (ON.2 - O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF). Funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 609551, SyncFree project.

3. REFERENCES

[1] P. Bailis and A. Ghodsi. Eventual consistency today: Limitations, extensions, and beyond. *Queue*, 11(3):20:20–20:32, Mar. 2013.

[2] K. P. Birman and T. A. Joseph. Reliable communication in the presence of failures. *Trans. on Computer Systems*, 5(1):47–76, Feb. 1987.

[3] S. Cribbs and R. Brown. Data structures in Riak. In *Riak Conference (RICON)*, San Francisco, CA, USA, oct 2012.

[4] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: Amazon's highly available key-value store. In *Symp. on Op. Sys. Principles (SOSP)*, volume 41 of *Operating Systems Review*, pages 205–220, Stevenson, Washington, USA, Oct. 2007. Assoc. for Computing Machinery.

[5] S. Gilbert and N. Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, 2002.

[6] R. A. Golding. *Weak-consistency group communication and membership*. PhD thesis, University of California Santa Cruz, Santa Cruz, CA, USA, Dec. 1992. Tech. Report no. UCSC-CRL-92-52.

[7] P. Mahajan, L. Alvisi, and M. Dahlin. Consistency, availability, and convergence. Technical Report UTCS TR-11-22, Dept. of Comp. Sc., The U. of Texas at Austin, Austin, TX, USA, 2011.

[8] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski. A comprehensive study of Convergent and Commutative Replicated Data Types. Rapp. Rech. 7506, Institut National de la Recherche en Informatique et Automatique (INRIA), Rocquencourt, France, Jan. 2011.

[9] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski. Conflict-free replicated data types. In X. Défago, F. Petit, and V. Villain, editors, *Int. Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS)*, volume 6976 of *Lecture Notes in Comp. Sc.*, pages 386–400, Grenoble, France, Oct. 2011. Springer-Verlag.

[10] W. Vogels. Eventually consistent. *ACM Queue*, 6(6):14–19, Oct. 2008.