Data Replication for Distributed Emergency Management Systems

Øyvind Hanssen[1]

[1] UiT Arctic University of Norway, Tromsø, Norway
oyvind.hanssen@uit.no

Abstract. This work explores how existing, potentially autonomous applications for crisis management and emergency response can be extended with replication and where nodes are mobile, offline or weakly connected.

Replication with strong eventual consistency, using operation-based CRDT, specifically the (LWW-Element-Set) is tested and evaluated.

Keywords: Crisis management, offline, replication, operation based CRDT.

1 Motivation and background

Emergency management operations may span large geographical areas and involve decentralised organisations, operating with mobile units and ad-hoc infrastructure. We may need to deal with situations when nodes become intermittently offline or experience weak connectivity and where conventional client-server architectures struggle.

In such settings, a *local-first* approach seems appropriate where each node can continue operating autonomously and later synchronize its state when connectivity is re-established.

We focus on server-applications backed by database systems, possibly running on portable devices and offering services on a LAN. These may originally be designed to be stand-alone but extended to share and replicate data to other application instances. For instance in [1] required data types in a decision support system for search and rescue include incidents, missions, personnel, vehicles, logs, etc. The goal is *eventual consistency* where the state of nodes may differ temporarily but converge to a global consistent state over time.

2 Research approach

This work retrofits an existing geographic information system (used by volunteer rescue services) with a robust replication layer that tolerates network partitions while preserving data integrity. How to do this efficiently and how to avoid or resolve conflicts resulting from concurrent updates are well known research questions. Work on Conflict-Free Replicated Data Types (CRDTs) [2,3] is taking this in a promising direction. It is investigated how CRDTs can be applied in this context.

The approach is to start with operation-based, specifically a Last-Writer-Wins (LWW) Element-Set CRDTs and see how this can provide *strong eventual consistency* for mutable datasets in a hierarchical overlay network. Nodes form a subscription graph (tree) where a child node subscribes to parent's datasets and where subscriptions persist across disconnections.

3 Design and implementation

Each dataset is modeled as a set of elements identified by immutable UUIDs. Operations are: ADD, UPD and DEL to insert, replace or remove an element respectively. Each operaton-message carries a timestamp based on the system clock. To resolve potential conflicts the LWW policy selects the operation with the latest timestamp. This ensures that operations on he same UUID are commutative, and the system state will eventually converge regardless of the message arrival order. Some edge-cases that stem from concurrent invocations need special attention. E.g. a DEL followed by an UPD. These cases don't make sense in a causal order but can occur here.

In the implementation, a *replication layer* is responsible for mapping local updates to outgoing messages representing operations, for mapping incoming messages to calls to the dataset implementations and for propagating operations to other nodes, etc. Metadata and outgoing messages are handled by a relational database. The implementation also supports detection of *causal stability* (see e.g. [3]) by assuming that messages are delivered in a causal order and by using ACK messages.

4 Discussion

Experiments confirm that replicas converge to identical states after reconnection. Operation-based messages remain lightweight compared with state-based CRDTs, yielding near-real-time updates when connectivity permits. The design also aligns nicely with existing RESTful services.

Reliance on synchronized physical clocks can be fragile in highly mobile or isolated environments and may need some attention.

The study underscores a trade-off between simplicity (LWW-Element-Set) and more expressive conflict handling. The LWW policy may discard legitimate concurrent edits or deleted elements may re-appear, suggesting exploration of "*observed-re-move*" or "*add-wins*" semantics (see e.g. [3]). Leveraging causal information, here by detecting causal stability could enable richer policies which more precisely represent user's intent. This work may be a base for looking into more complex cases.

References

1. Holme, Jarl. "Exploring the viability of a CRDT-based Data Model for a distributed decision support system", Master thesis, NTNU, 2020, https://hdl.handle.net/11250/2656668.
2. Shapiro, M., Preguiça, N., Baquero, C., Zawirski, M. (2011). Conflict-Free Replicated Data Types. In: Défago, X., Petit, F., Villain, V. (eds) Stabilization, Safety, and Security of Distributed Systems. SSS 2011. Lecture Notes in Computer Science, vol 6976. Springer.
3. Paulo Sérgio Almeida, Approaches to Conflict-free Replicated Data Types. ACM Computing Surveys. 57, 2, Article 51 (February 2025), 36 pages. https://doi.org/10.1145/3695249