Dynamic Data > Challenges

Garbage Collection

How does a replica safely compact CmRDT history?

GitHub, Inc. github.com/protocol/research/issues/9

This repository Search Pull requests Issues Marketplace Explore

protocol / research

Watch 30 Star 16 Fork 3

Code Issues 7 Pull requests 1 Insights

Optimize storage and convergence time in causal CmRDTs #9

Edit New issue

Open pgte opened this issue 6 days ago · 0 comments

pgte commented 6 days ago • edited by miyazono Member

Optimize storage and convergence time in causal CmRDTs

Context

Eventual consistency

Unlike Strongly Consistent systems, Eventually Consistent (EC) systems don't require synchronization between peers in order to modify data. Changes can be done locally or to a small number of replicas and then asynchronously replicated to others, eventually reaching them. These systems are more resistant to network partitions, and thus are suited to being used in decentralized environments where connectivity can be low. The drawback is that, in a given point in time, data is not guaranteed to be synchronized between peers.

Strong Eventual Consistency

Strong Eventual Consistency (SEC) is a stronger type of EC where some properties are guaranteed: when all the replicas have received all the messages independently of their order, they are guaranteed to reach the same state.

Assignees

No one assigned

Labels

open problem statement

Milestone

No milestone

Notifications

Unsubscribe

You're receiving notifications because you authored the thread.

2 participants