Real-Time Collaborative Text Editor

Vasu Singhal, Tanish Lad, Daksh Rawat, Parv Joshi

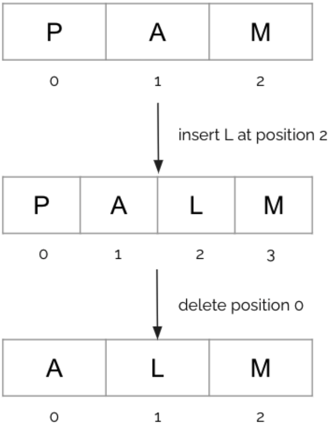
Distributed Systems Project: Team 21

Table of Contents

Table of Contents

What is a Text Editor?

A text editor is a space where you can insert or delete text characters and then save the resulting text to a file. Each character has a value and a numerical index that determines its position in the document.

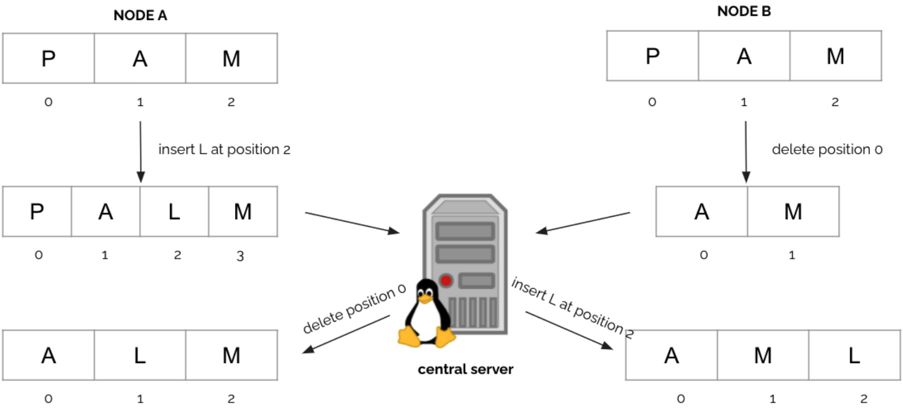


(Real-Time) Collaborative Text Editor

- One user editing a document seems simple, but what if we want multiple users simultaneously editing the same document? How would that work?
- A collaborative text editor is a text editor that allows multiple users located at different locations, to share a document among themselves and edit it. The edits made by any user must be reflected back to each user sharing the document.
- Popular examples of such (real-time) text editors include Google Docs, Overleaf, Office Online, etc. When you type in a Google Doc, the keystrokes are immediately applied to the local copy of the document in your web browser, without waiting for them to sync to a server or any other users. This means that when two users are typing concurrently, their documents can temporarily diverge; as network communication takes place, the system needs to ensure that all users converge to the same view of the document.
- An example of a non-real-time collaborative editor is a version control system like Git.

Why is this hard?!

A naive approach to build such an editor might look something like this:



Why is this hard?!

- Oh No! One user has "ALM" and the other user has "AML". It's not hard to see what went wrong!
- They did not converge to the same state. This demonstrates one of the primary challenges with building a collaborative text editor — getting all users to converge to the same document state.
- The insert and delete operations were applied in different orders i.e. the operations did not commute. Operations should also follow associativity.
- Idempotency: If the same operation is multiple times, then the result must be same.
- We need a data structure which follows the above property.
- The end goal is that all the nodes must converge to the same state. But same state is not enough! We also should make sure that there is no data lost.

Table of Contents

Algorithms for Convergence

1. OT

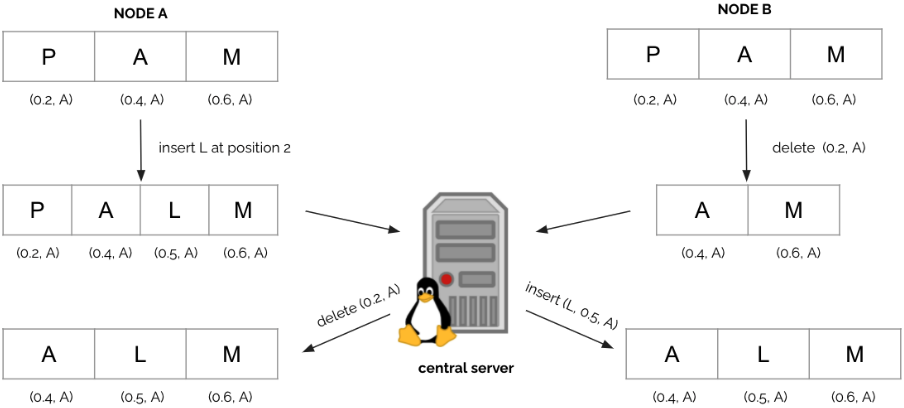
- Operational Transformation (OT) is an algorithm that compares concurrent operations and detects if they will cause the documents to not converge. If the answer is yes, the operations are modified (or transformed) before being applied.
- OT was the first popular way to allow for collaborative editing. Unfortunately, it's extremely tough to actually implement.

2. CRDT

- A conflict-free replicated data type (CRDT) is a data structure which can be replicated across multiple computers in a network, where the replicas can be updated independently and concurrently without coordination between the replicas, and where it is always mathematically possible to resolve inconsistencies that might come up.

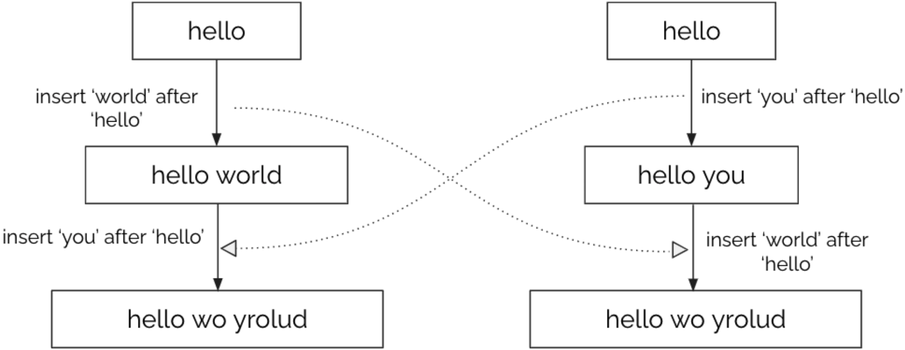
Text Editor using CRDT

Properties are added to each character object that enable commutativity and idempotency. For e.g. each character this time is given a unique identifier that does not depend on the neighbouring characters and does not change with time.

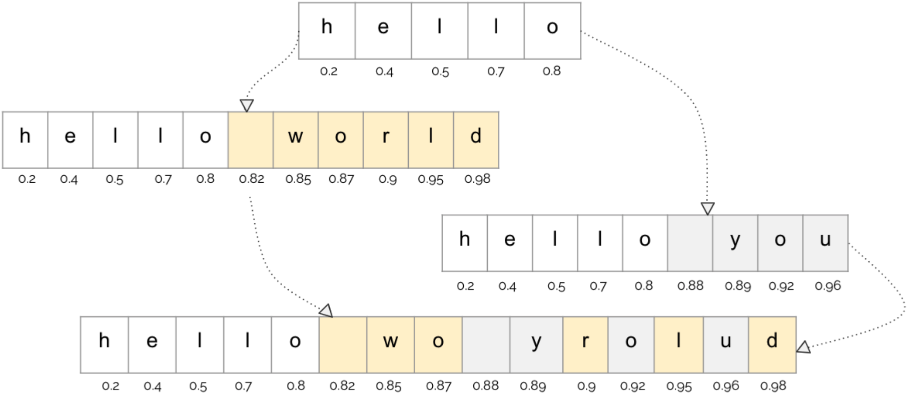


Interleaving Anomalies

Two concurrent insertions at the same position might get interleaved

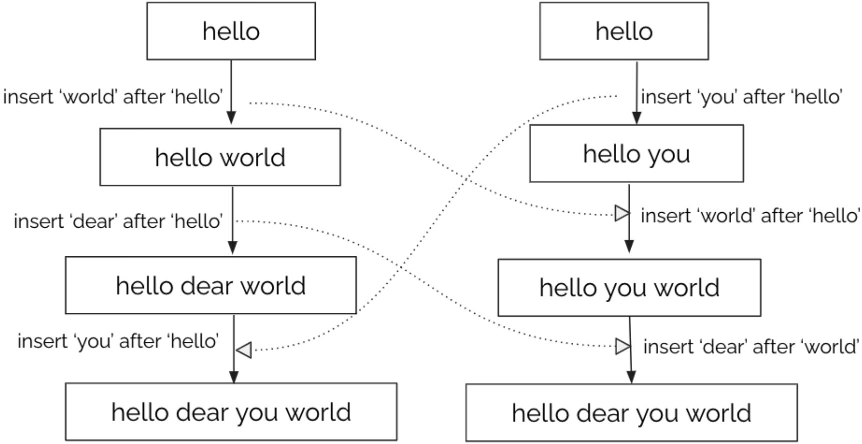


Why did this happen?



RGA Algorithm

A lesser interleaving anomaly: no interleaving at character level, but interleaving at world level is still possible.



RGA Algorithm

The final text is formed by doing a depth first traversal of the tree.

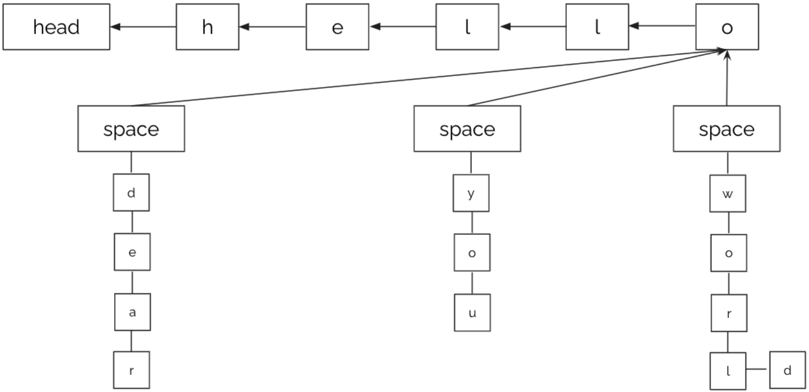


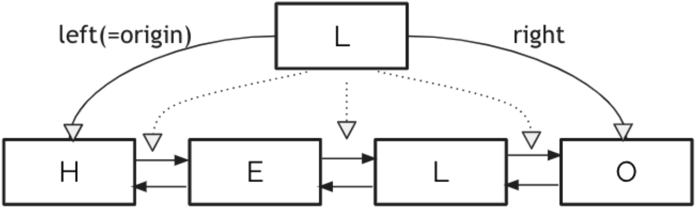
Figure: The tree structure underlying the RGA: each node represents a character, and its parent is the immediate predecessor character at the time it was inserted. The document state corresponds to a depth-first pre-order traversal over this tree, with sibling nodes visited in descending timestamp order

YATA Algorithm

Each node is represented as the following:

id	*origin*	*left*	*right*	*Is Deleted*	*content*

A total ordering of all the nodes based on its structure ensures that conflicts are resolved in the same way by all processes.



Performance Comparison

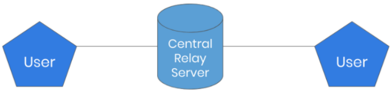
CRDT	Local INS	Local DEL	Remote INS	Remote Del
RGA	O(H)	O(H)	O(H)	O(log(H))
YATA	O(log(H))	O(log(H))	O(H^2)	O(log(H))

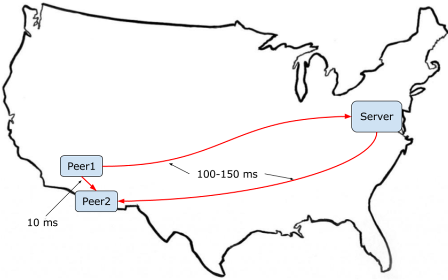
Table: Performance Comparison between RGA and YATA

Table of Contents

Limitations of a Central Relay Server

- High Latency between users
- Scaling requires additional resources and money
- Must trust the central server
- Single point-of-failure





WebRTC

- We can remove these limitations by switching to a peer-to-peer architecture where users send operations directly to each other. In a peer-to-peer system, rather than having one server and many clients, each user (or peer) can act as both a client and a server.
- To allow nodes to send and receive messages, we used a technology called WebRTC. WebRTC is a protocol that was designed for real-time communication over peer-to-peer connections.
- It's primarily intended to support plugin-free audio or video calling but its simplicity makes it perfect for us even though we're really just sending text messages.
- While WebRTC enables our users to talk directly to one another, a small server is required to initiate those peer-to-peer connections in a process called "signaling".

Table of Contents

The System

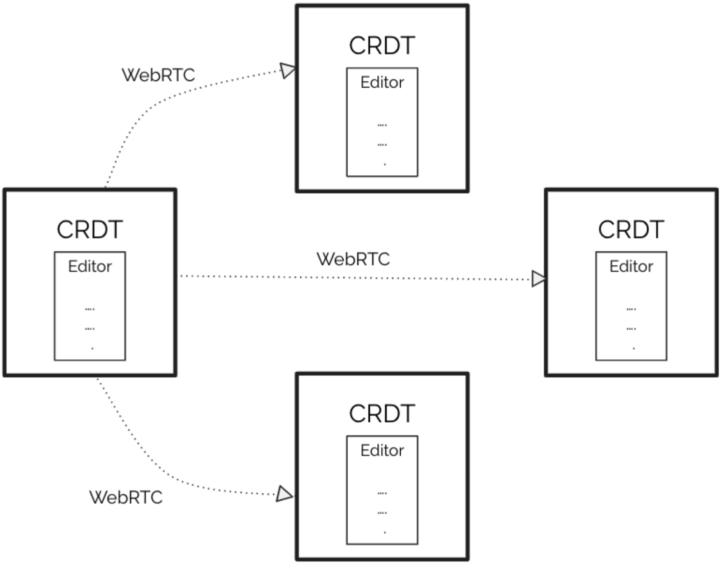

CRDT
Editor
WebRTC
CRDT
Editor
WebRTC
CRDT
Editor
CRDT
Editor
WebRTC

Table of Contents