CS 7610: Final Project

NUpad using Conflict-Free Replicated JSON Datatype

The one where we don't store your data

Sumeet Mahendra Gajjar Madhur Akshay Jain

Original Paper: A Conflict-Free Replicated JSON Datatype

Proposal statement: In this project, we want to implement a algorithm to convert a JSON object into a CRDT Datatype. We will initially start with a **List** data structure and later will extend the project to implement a **Map** data structure. The data structure will automatically resolve concurrent modifications such that no updates are lost, and such that all replicas converge towards the same state upon receiving all the messages eventually.

Using this datatype, we would create a collaborative editing application similar to Google Docs. As of now, the application will have following components:

1. **CRDT Client Library**: This is the crucial component of the project. It will contain the core logic to merge the various updates coming from peers and emit a consistent JSON object.
2. **Central Multicast Server**: All peers will send their updates to the server and the updates will be multicasted to all the peers. The peers will then use the CRDT Client library to apply the updates to the doc.
3. **Bare minimum UI**: The UI will be used for editing the doc and displaying the merged contents.
4. **End to End encryption (Stretch Goal)**: In our design for Google docs, no data is stored with a central server, all the data is stored on clients. The role of the server here is to just propagate the updates to all peers. This design choice helps us implement an actual end-to-end encryption with promise of no data being stored at server. If time permits, we would like to implement this.
5. **Replace Central server with a P2P network (Stretch Goal)**: As stated multiple times, the server's role is just to propagate updates. So it can be easily replaced with a decentralized P2P network to achieve further anonymity. If time permits, we would like to implement this.