Efficient Approach to Move Elements in a Geo-Replicated Tree

Parwat Singh Anjana[1] Adithya Rajesh Chandrassery[2] Sathya Peri[1]

[1]*Indian Institute of Technology, Hyderabad, India* [2]National Institute of Technology Karnataka, Surathkal, India

OBJECTIVE

Replicated tree data structures are extensively used in collaborative applications and distributed file systems, where clients often perform move operations.

We present an efficient algorithm to perform move operations on the distributed replicated tree while ensuring eventual consistency and maintaining the tree structure.

MOTIVATION AND CHALLENGES

- It is difficult to implement move operations on the replicated trees because concurrent operations by multiple clients may result in cycles, as shown in Figure 1.
- Popular file systems such as Dropbox and Google Drive also face this problem.
- Dropbox duplicates the nodes with conflict to avoid cycles whereas Google Drive allow only one of the clients to be synchronized with the server while the other one can possibly be in an error state.
- Ensuring correctness while providing high availability can be challenging.
- Kleppmann et al. [2] proposed an approach that undoes and redoes operations based on timestamps and ensure strong eventual consistency.
- Kleppmann's approach is computation-intensive, requires many compensation operations.

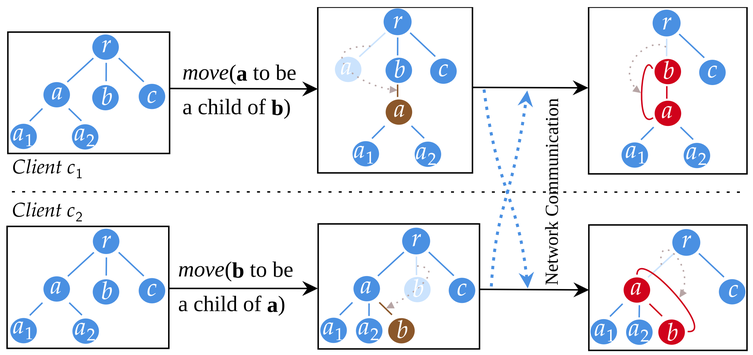

Figure 1. Difficulty with the move operation on a replicated tree.

HIGHLIGHTS

- The proposed technique is primarily concerned with resolving conflicts efficiently and works well with network partitions.
- The proposed algorithm guarantees strong eventual consistency to show the *convergence* of the replicas and the maintenance of the tree structure; correctness proofs are provided in technical report [1].
- We achieved an effective speedup between 14.6× to 68.19× over Kleppmann's approach for the remote move operations on a distributed replicated tree.

REFERENCES

[1] Parwat Singh Anjana, Adithya Rajesh Chandrassery, and Sathya Peri. An efficient approach to move elements in a distributed geo-replicated tree. *arXiv preprint arXiv:2203.10285*, 2022.

[2] Martin Kleppmann, Dominic P. Mulligan, Victor B. F. Gomes, and Alastair Beresford. A highly-available move operation for replicated trees. *IEEE Transactions on Parallel and Distributed Systems*, pages 1–1, 2021.

PROPOSED METHODOLOGY

- The data-structure considered supports three operations
  - *Inserting* a new node in the tree.
  - *Deleting* a node from the tree.
  - *Moving* a node along with a sub-tree to a child of a new parent in the tree.
- These three operations can be represented as changing the parent of the node. An operation is represented as - *ts (timestamp), n (node to be moved), p (new parent of n)*.
- Before applying an operation the algorithm checks if the result causes a cycle or not.
- Consider a move from *n* to *p*. If node *n* is an ancestor of node *p* then it can lead be a potential cycle. In this case, the node with the latest timestamp in this range will be moved back to one of its safe previous parents in order to prevent the cycle, as shown in Figure 3.
- A fixed number of previous parents for each node is stored in *log*.
- In case none of the previous parents are safe, then the node is moved to special node called *CONFLICT*.
- Deleted nodes are attached to a special node called *TRASH*.
- This compensation operation now has to be propagated to other replicas.
- For each node, we store the timestamp of the last operation applied on it, and only apply a new operation if it's timestamp is greater than the last one, as shown in Figure 2.

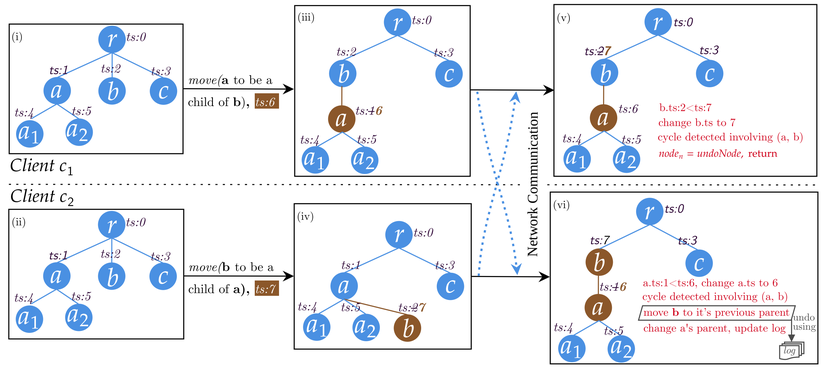

Figure 2. Preventing cycles in proposed approach.

EXPERIMENTAL EVALUATION

- Using Microsoft Azure E2s_v3 VM instances, the algorithm was executed on 3 replicas in different geographical locations.
- The performance of the proposed approach is compared against the Kleppmann et al. [2], as shown in Figure 4 and Figure 5.
- The proposed approach achieves an effective speedup between 14.6× to 68.19× over Kleppmann's approach for the remote move operations.
- In our approach the average time to apply remote move operations remains constant whereas in the Kleppmann's approach there is a steady increase. A big performance gap can be noticed between both the approaches in case of remote move operations.

CONCLUSION AND FUTURE WORK

- We proposed a novel algorithm for computationally efficient and low latency move operation on the replicated tree.
- The proposed technique requires a single compensating operation to undo the effect of the cyclic operation.
- Identifying the optimal number of previous parents and implementing an efficient move operation on other CRDTs is left as future work.

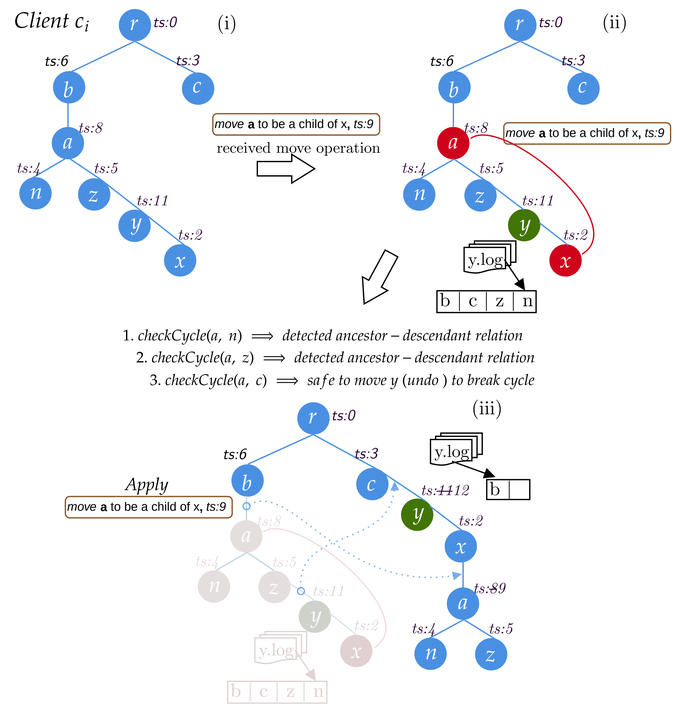

Figure 3. Remote move operation handling at a replica.

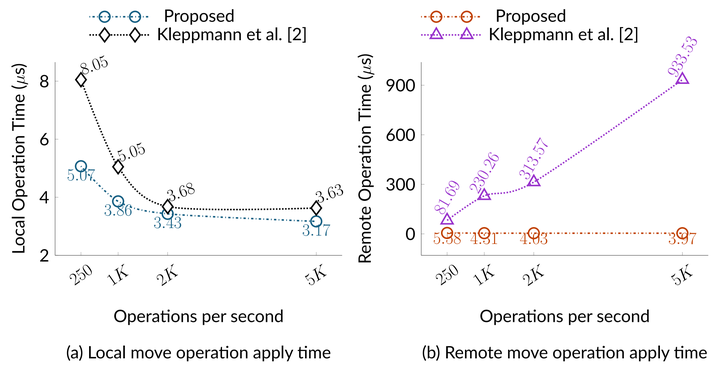

Figure 4. Time to apply a move operation with varying operations.

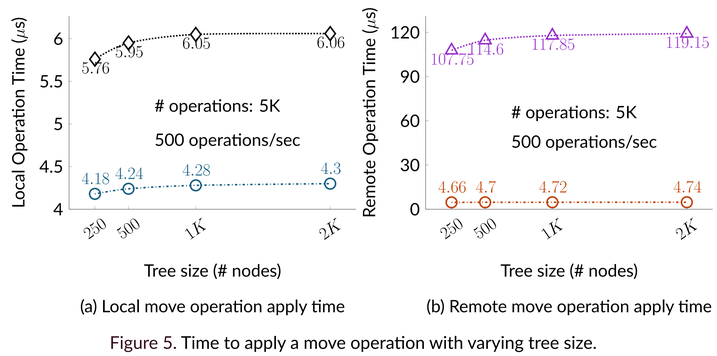

Figure 5. Time to apply a move operation with varying tree size.

The full version of this work got accepted at the IEEE International Conference on Cloud Computing (CLOUD), Barcelona, Spain, July 11-16, 2022

mailto: cs17resch11004@iith.ac.in, adithyarajesh.191cs203@nitk.edu.in, sathya_p@cse.iith.ac.in