

Efficient (re) naming in Conflict-free Replicated Data Types (CRDTs)

Matthieu Nicolas
matthieu.nicolas@inria.fr

Inria, F-54600, Université de Lorraine, LORIA, F-54506, CNRS, F-54506

- Abstract (maximum of 100 words).
- Clear statement of the identified research problem(s) and the context in which the problem(s) will be addressed.
 - Distributed applications need to make a trade-off between availability and consistency
 - To ensure high availability, adopt the optimistic replication model paired with the eventual consistency model to replicate data
 - A conflict resolution mechanism is needed to solve concurrent updates
 - The CRDT approach
 - To converge, some CRDTs attach identifiers to each element of the data structure
 - Identifiers' size grows over time, decreasing application's performance
 - CRDTs representing replicable sequences are some of the ones suffering the identifiers' growth issue
- Summary (with appropriate references) of the state-of-the-art related to the identified problem(s) along with a clear restatement of the “gap” relative to the research problem(s).
 - A sequence represents a number of ordered values
 - Can insert or remove an element at a position
 - Several CRDTs allow to replicate sequences : Treedoc, Logoot, LogootSplit
 - To ensure convergence, attach an identifier to each inserted element
 - Identifiers allow to identify an element and order them
 - Thus identifiers have to comply to several constraints : globally unique, totally ordered and belong to dense set
 - Because of these constraints, grow quicker
 - In Core and Nebula, authors propose mechanism to rename identifiers to limit growing rate
 - Proposed mechanism requires consensus and to determine a "core" of stable and durable nodes
 - Want to address this issue in a fully distributed manner, without any kind of super-peers
- Statement explaining the approach and results, according to PhD stage: state the intended approach including a summary of work accomplished to date (if any).
 - Defined properties that renaming functions need to comply to (commutative with insert/delete, a rename can be undone, do not shuffle the order)
 - Designed corresponding functions

- Designed a new version of LogootSplit integrating them
- Description of evaluation : describe the evaluation plan including intended metrics (quantitative and/or qualitative).
 - Implementing this new CRDT into MUTE
 - Testing it in every scenario we were able to think of
 - Need to prove the correctness of these renaming functions
 - Taking a look to model checking and automatic theorem provers
 - In parallel, designing a benchmark for collaborative editing tools (NOTE: mauvais terme ?)
 - Goal is be able to replay a collaborative editing session from its logs using different conflicts resolution mechanisms
 - Intend to ensure that the implementation is correct
 - Intend to mesure the performances (memory, cpu usage and bandwith usage, time to integrate an update, time required to converge)
- Conclusion: that includes a statement of the real or potential impact(s) of solving the identified research problem(s). Finishers especially should include a brief conjecture about future work that builds on the PhD research.
- Acknowledgements that properly recognize others' contributions to the work (supervisor(s), other graduate students, funding sources, etc.).