Paper resume « Evaluating CRDTs for Real-time Document Editing »

Ronan-Alexandre Cherrueau
Adrien Bougouin
Alban Ménager

Year 2011-2012
(University of Nantes)

Nowadays, colaboring work is increasing and real-time editing systems are catching on. Google Docs is a good example, it enable multiple authors to edit the same document at the same time, whatever there location or the time in each location.

For now, real-time editing systems use a replication mechanism to ensure consistency when merging concurrent changes performed on the same document. Optimistic replication gives to users a low time of latency. But they use centralise approaches and the probleme with this is that it's not only storing documents information but personal informations too and it could be a privacy threat if they are used by corporations.

So, to prevent this, a good solution for not using centralised mechanism is to replace it by an decentralised one. Peer-to-peer seems to be a solution. But, the main key factor is performance. Obviously, if the editing application can't respond to user action in a reasonable time (about 50ms), he may get frustrated and leave the current application.

The goal of this experiments is to select some algorithms based on optimistic replication and evaluate them on a decentralised real-time collaborative editing systems. The evaluations will be based on real context on the sames conditions and using the same data flow.

The first approache is *Operation Transformation*. This approche take in count the effect of concurrents operations. An *Operation Transformation* is used to keep document consistency after severals concurrents operations. Google Docs use an algorithm named *Jupiter* and user vector clock to detect concurrents operations, but this solution not scale well in peer-to-peer system.

A new approache called *Commutative Replicated Data Types* (CRDT) for peer-to-peer environment was introduced as a new class of replication mechanisms to preserve consistency. CRDT doesn't required concurrent operations detections because it's designed for concurrent operations to be natively commutative.

CRDT have some caracteristic :
- The concurrent operations are natively commutative.
- The document is a linear sequence of elements.
- A unic position identifier.

For the experiments, they select differents algorithms for generating the unic position identifier :
- Logoot
- RGA
- WOOT
- WOOTO
- WOOTH

Figure 1 p.2 (with R the number of replicas and by H the number of operations on the document) shows the theoretical evaluation of this differents algorithms. RGA and Logoot have the bests results.

ALGORITHM	LOCAL		REMOTE	
	INS	DEL	INS	DEL
WOOT	$O(H^3)$	$O(H)$	$O(H^3)$	$O(H)$
WOOTO	$O(H^2)$	$O(H)$	$O(H^2)$	$O(H)$
WOOTH	$O(H^2)$	$O(H)$	$O(H^2)$	$O(log(H))$
Logoot	$O(H)$	$O(1)$	$O(H.log(H))$	$O(H.log(H))$
RGA	$O(H)$	$O(H)$	$O(H)$	$O(log(H))$
SOCT2/TTF	$O(H+R)$	$O(H+R)$	$O(H^2)$	$O(H^2)$

FIGURE 1 – Worst-case time-complexity analysis

But this is just theoretical. The equip of the paper designed real-time peer-to-peer collaborations in order to obtain there logs and apply them on the algorithms. For real applications, two experiments and severals groups have been mades :

- 3 grups have to do there semester report only using the collaborating editor during one and a half hours :
  - 2 grups of 4 students.
  - 1 grup of 5 students.
- 9 grups of 2 students have to translate an episode of *The Big Bang Theory*

Figure 2 p.3 shows the total number of users/character operations :

- A user operation is adding/deleting letters or grups (copy/past).
- A character operaton is the transformation of each user operation into a character operation (for example, copy/past "alma" are the character operations :
  - adding 'a' at space 0
  - adding 'l' at space 1
  - ...

	Report			Series	
	group 1	group 2	group 3	doc 1	doc 2
No. user operations	11 211	11 066	13 702	9 042	9 828
No. char. operations	26 956	47 992	42 443	29 882	10 268
% of del	12	12	12	9	5

Figure 2 – Total number of user/character operations

All the algorithms have been runned 10 times with same datas and sames conditions. Figures 3 and 4 p.3 shows the total number of users operations.

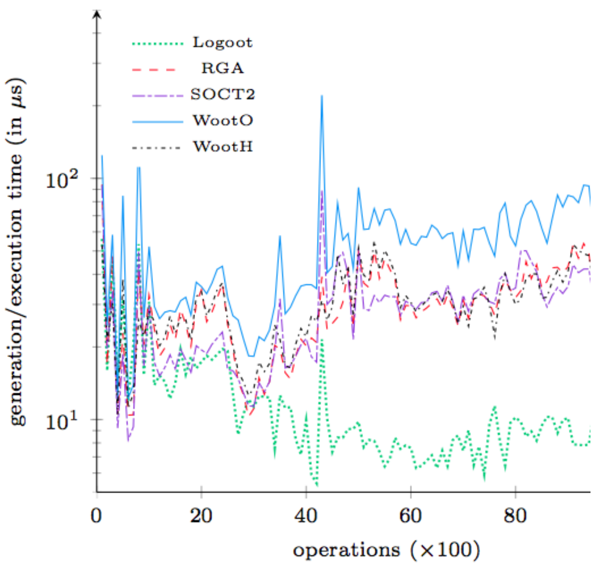

Figure 3 – User operation execution times - 2nd group report

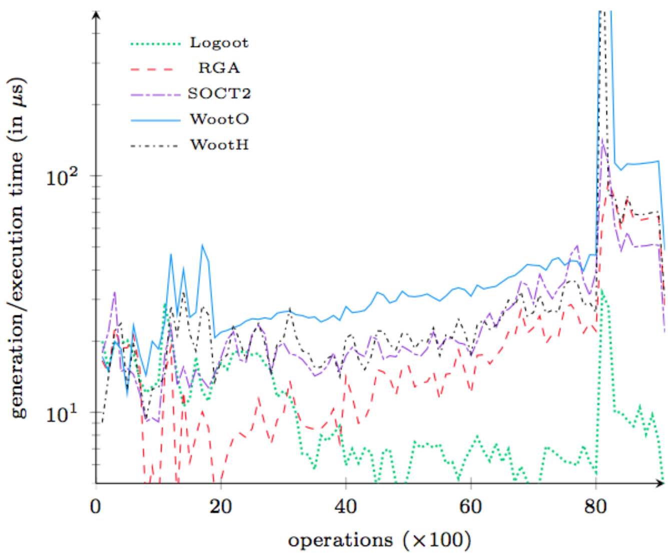

Figure 4 – User operation execution times - 1st series

The increases are due to insert or delete a large group of characters on the document. All the algorithms decrease over the time because of the size of the document (the unique identifier is bigger when there is a lot of operations) but logoot decrease slowly than the others.

Figures 5 and 6 p.4 shows the total number of characters operations.

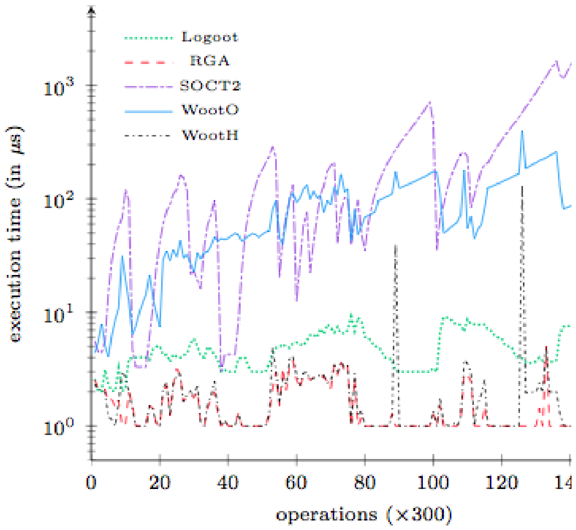

FIGURE 5 – Character operation execution times - 2nd group report

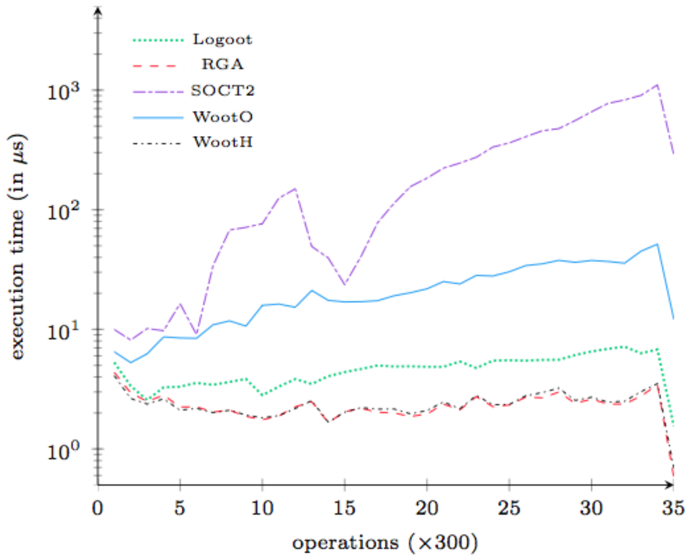

FIGURE 6 – Character operation execution times - 1 time series

Wooto, Logoot and RGA remains stable during al the time. The behavior of SOCT2 performance is mainly due to its garbage collection mechanism. Users had a period of inactivity and the garbage mechanism cannot purge the history log. SOCT2 can't be used for real-time editing because of his response time increasing 50ms quickly.

Figure 5 is a bite different than 6 because of the number of characters deletes (students didn't delete much less in the series experiments as in the report experiment). RGA and WootH are better than the other because there algorithms are based on hash tables.

In conclusion, this is a first the first performance evaluation of algorithms with real collaboration traces including concurrency. This experimentation demonstrate the suitability of CRDT algorithms in real-time collaboration and they outperform some representative opera- tional transformation approaches that were well established for real-time collaboration in terms of local generation time and remote integration time.