INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Logoot: a P2P collaborative editing system

Stéphane Weiss — Pascal Urso — Pascal Molli

N° 6713

octobre 2008

Thème COG

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6713--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche NANCY - GRAND EST

Logoot: a P2P collaborative editing system

Stéphane Weiss *, Pascal Urso * , Pascal Molli *

Thème COG — Systèmes cognitifs
Équipes-Projets ECOO

Rapport de recherche n° 6713 — octobre 2008 — 13 pages

Abstract:
Massive collaborative editing becomes a reality through leading projects such as the Wikipedia. Such massive collaboration is currently supported with costly central service. To avoid such costs, we aim to provide a peer-to-peer collaborative editing system. Existing approaches that propose distributed collaborative distributed either do not scale in term of users number or in term of editions number.

We present the Logoot approach that scales in these both dimensions while ensuring causality, consistency and intention criteria. We evaluate the Logoot approach and compare it to others with a corpus of all the editions applied on a set of the most edited and biggest page of the Wikipedia.

Key-words: Logoot

* Loria - ECOO

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Édition collaborative pair-à-pair

Résumé :
L'édition collaborative massive devient une réalité à travers de projets telle que la Wikipedia. Une telle collaboration massive est pour l'instant supportées à l'aide de service centraux couteux. Afin d'éviter ces coûts, nous essayons de fournir un syst'eme d'édition collaborative pair-'a-pair. Les approches existantes proposant une édition collaborative distribuée ne passent pas à l'échelle en terme de nombre d'éditions.

Nous présentons l'approche Logoot qui passe à l'échelle dans ces deux dimensions tout en assurant les critères de causalité, de cohérence et d'intention. Nous évaluons l'approche Logoot et la comparons aux autres approches à l'aide d'un corpus d'éditions appliquées sur les plus grandes et les plus éditées pages de la Wikipedia.

Mots-clés : Logoot

1 Introduction

Collaborative editing (CE) systems allow distant users to modify the same data concurrently. The major benefits are: reducing task completion time, getting different viewpoints, etc ... Wiki and DVCS (distributed version control) systems are the most popular collaborative editing tools.

Several collaborative editing systems are becoming massive: they support a huge number of users to obtain quickly a huge amount of data. For instance, the Wikipedia which is edited by 7,5 millions of users, holds 10 millions of articles in only 6 years.

However, most of CE systems are centralized, hence, their scalability is costly. To avoid such costs, some prior works aim to provide a peer-to-peer (P2P) CE system. Moreover, P2P architectures also offers several other benefits: failure tolerance, resistance to censorship as well as adhoc collaboration and offline work.

Prior works on P2P CE [1, 2, 3, 4] are based on “tombstones”: a deleted line is replaced by a tombstone instead of removing it from the document model. In these approaches, tombstones cannot be directly removed without compromising the document’s consistency. Therefore, the overhead required to manage the document grows continuously.

This cost is not acceptable on massive editing systems. For instance, for the the most edited pages of Wikipedia[1] on a tombstone based system, the storage overhead often represents more than 100 times the document size.

Tombstone are also responsible of performance degradation. Indeed, in all existing approaches, the execution time of modification integration depends on the whole document size – including tombstones. Therefore, letting the tombstone number growing degrades the performance.

Therefore, we propose a novel approach called Logoot which does not require the use of tombstone. Moreover, the time complexity of Logoot is only logarithmic according to the document size. The Logoot approach correctness is based on the CCI criteria : Causality, Convergence and Intention.

A part of the contribution presented in this paper is a profiling of CE performance in “real condition”. Given a set of the most edited and the biggest pages of the Wikipedia, and thanks to Wikimedia API, we extract the list of all edits done on these pages since their creation. We have replayed these edits on our framework and in tombstone based frameworks. In Section 5, we show and analyse the results of this experiment.

2 P2P Collaborative Editing System

We make the following assumptions about P2P Collaborative Editing System (P2P CE).

A P2P CE network is composed by an unbounded set of peers. Each peer has the same role. A user is supposed working on each peer. Peers can enter and leave the network arbitrary fast. As a consequence, mechanisms as consensus or state vectors are not usable in this context. Each peer possesses a unique site identifier.

Each user hosts a replica of the document and can modify it at any time. Local modifications are eventually received on all other peers. We make no assumption about the propagation time. When a peer receives a modification, it replays it in its local replica. This kind of replication is known as optimistic replication [5] (or lazy replication).

[1] http://en.Wikipedia.org/wiki/Wikipedia:Most_frequently_edited_articles

The dissemination mechanism of the local modification through the P2P network is not detailed on this report. However, it can be achieve, for instance, by a lightweight probabilistic broadcast [6] coupled to an anti-entropy mechanism [7].

According to [8], a collaborative system is considered as correct if it respects the CCI criteria:

Causality: This criterion ensures that all operations ordered by a precedence relation, in the sense of the Lamport's *happened-before* relation [9], will be executed in the same order on every copy.

Convergence: The system converges if all replicas are identical when the system is idle.

Intention: The expected effect of an operation should be observed on all replicas. A well-accepted definition of operations intention for textual documents is :

delete A line must be remove from the document if and only if it has been deleted on a peer.

insert A line inserted on a peer must appear on every peer. And, the order relation between the document's lines and a newly inserted line must be preserved on every peer (as long as these lines exist).

3 Related Work

WOOKI [2] is a P2P wiki system which is based on Woot [10]. The main idea of Woot is to treat collaborative document as a Hasse diagram that represents the order induced by the *insert* operations. Therefore, the Woot algorithm computes a linear extension of this diagram. WOOKI barely respects the CCI correction criteria. Indeed, the causality is replaced by preconditions. As a result, the happened-before relation can be violated in some cases. The convergence is ensured by the algorithm and by using tombstones.

TreeDoc [3] is a collaborative editing system which uses a binary tree to represent the document. Deleted lines are also kept as tombstones. The authors propose a kind of "2 phase commit" procedure to remove tombstones. Unfortunately, such procedure cannot be used in an open-network such as P2P environments. However, this approach proposes also an interesting general framework called Commutative Replicated Data Type (CRDT) to build distributed collaborative editors ensuring CCI criteria.

[4] proposes a distributed replication mechanism in the CRDT framework, that ensures the CCI criteria but using tombstones and vector clocks.

MOT2 [11] is P2P peer-wise reconciliation algorithm in the Operational Transformation (OT) approach[12]. Such algorithm assumes the existence of transformation functions satisfying some properties. To our best knowledge, the only transformation functions usable [13] with MOT2 are the Tombstone Transformation Functions (TTF) which are based on tombstones.

DTWiki is a disconnection tolerant wiki based on TierStore, itself based on Delay-Tolerant Network. The causality is achieve in DTWiki using Version Vectors. In DTWiki, the conflict resolution slightly differs from approaches like WOOKI or TreeDoc. Indeed, in case of concurrent modifications of the same wiki page, DTWiki generates one revision per concurrent modifications. Revisions are not automatically merged, even in case of non-conflicting modifications. On the contrary, all approaches considered above, automatically merged concurrent modifications.

Repliwiki is a P2P wiki system using the Summary Hash History approach to detect causality and concurrency. However, the authors do not provide any reconciliation mechanism.

Distriwiki is a P2P wiki system based on JXTA. Unfortunately, the authors do not discuss of the concurrent updates case.

4 Proposition

Our idea is based on the CRDT [3] framework for collaborative editing. In the CRDT framework, modifications produced locally have to be re-executed on remote sites but possibly in different orders. The main idea of this framework is to use a data type where all concurrent operations commute. Combined with the respect of the causality relationship between operations, this commutation ensures the convergence criteria.

To achieve commutativity on a linear structure, the authors propose a solution based on a total order between elements in the document. More precisely, there is two kinds of modification :

- $insert(pos, line)$ that inserts the $line$ content at the position pos.
- $delete(pos)$ that removes the line at the position pos.

In the original paper, a tree structure is introduced to maintain the total order between positions. However, safely removing elements on a tree can not be achieved without tombstones. In [4], authors refer also to the CRDT framework but use vector clocks to manage the order.

Our idea is to use a simple position identifier based on list of integers for each line. With such an identifier, a line can be removed from the document model without affecting the order of the remaining lines.

4.1 Logoot model

A Logoot document is composed by *lines* defined by: $\langle pid, content\rangle$ where $content$ is a text line and pid a unique *position identifier*. There is two virtual lines called l_B and l_E to represent the beginning and the ending of the document.

The main idea to insert a *line* is to generate a new position A such as $P \prec A \prec N$ where P is the position of the previous line and N the position of the next line.

$\langle pid_0, l_B\rangle$
$\langle pid_1$, "This is an example of a Logoot document" $\rangle$
$\langle pid_2$, "Here, $pid_1 \prec pid_2$ " $\rangle$
$\langle pid_3$, "And $pid_2 \prec pid_3$ " $\rangle$
$\langle pid_\infty, l_E\rangle$

To allow operations to commute, position identifiers must be unique. Also, since a user can always insert a line, we must be able to generate a position A such as $P \prec A \prec N$ for any P and N.

In the following definition we assume that each site maintains a persistent logical clock $clock_s$ incremented each time a line is created.

Definition 1. • *An* identifier *is a couple $\langle p_i, s_i\rangle$ where p_i is an integer and s_i a site identifier.*

- *A* position *is a list of identifier.*
- *A* line identifier *generated by a replica s is a couple (pos, h_s) where $pos = i_0.i_1.\ldots.i_{n-1}.\langle p, s\rangle$ is a position and h_s is the value of $clock_s$.*

Thus, every line identifier is unique since the last identifier of the list $i_0.i_1.\ldots.i_{n-1}.\langle p, s\rangle$ contains the unique site identifier and the value of the logical clock of this site.

To obtain a total order between positions, we use the following definition.

Definition 2.
- *Let $p = p_0.p_1 \ldots p_n$ and $q = q_0.q_1 \ldots q_m$ be two positions, we get $p \prec q$ if and only if $\exists j \leq m.\ (\forall i < j.\ p_i = q_i) \wedge (j = n + 1 \vee p_j <_{id} q_j)$*
- *Let $id_1 = \langle int_1, s_1\rangle$ and $id_2 = \langle int_2, s_2\rangle$ be two identifiers, we get $p_1 <_{id} p_2$ if and only if $int_1 < int_2$ or if $int_1 = int_2$ and $s_1 < s_2$.*

We only compare positions – and not logical clocks – since there can not be, in the same model, two lines with the same position.

Finally, a Logoot document looks like:

$\langle\langle(\langle 0,0\rangle), NA\rangle, l_B\rangle$
$\langle\langle(\langle 1,1\rangle), 0\rangle$, "This is an example of a Logoot document" $\rangle$
$\langle\langle(\langle 1,1\rangle.\langle 1,5\rangle), 23\rangle$, "The replica on site 5 find a place between 1 and 1" $\rangle$
$\langle\langle(\langle 1,3\rangle), 2\rangle$, "This line was the third made on replica 3" $\rangle$
$\langle\langle(\langle MAXINT, 0\rangle), NA\rangle, l_E\rangle$

4.2 Modifying a Logoot document

To insert a $line$, we have to generate a position p such as $p_1 \prec p \prec p_2$ with p_1 the position of the previous line and p_2 the position of the following line.

To generate this position, we use the following definition:

Definition 3. *On a site which identifier is s, a line inserted between a position $p = p_0 \ldots p_n$ and a position $q = q_0 \ldots q_m$ will have a shortest position $r = p_0 \ldots p_i.\langle x, s\rangle$ such that $p \prec r \prec q$.*

For instance, an insertion position

- between $\langle(\langle 2,4\rangle), 3\rangle$ and $\langle(\langle 10,5\rangle), 6\rangle$ is $\langle(\langle x,s\rangle), h_s\rangle$ with $x \in]2, 10[$
- between $\langle(\langle 5,1\rangle.\langle 1,6\rangle.\langle 10,2\rangle), 54\rangle$ and $\langle(\langle 5,1\rangle.\langle 2,1\rangle.\langle 15,2\rangle), 23\rangle$ is $\langle(\langle 5,1\rangle.\langle 1,s\rangle), h_s\rangle$ if $s > 6$; else $\langle(\langle 5,1\rangle.\langle 1,6\rangle.\langle x,s\rangle), h_s\rangle$ with $x \in]10, MAXINT[$.

To choose the integer x any arbitrary choice can be made. However, to restrain two different sites to generate concurrently the same choice, and thus to reduce the grow rate of position list, we apply a random function.

4.3 Integrating remote modifications

Both line insertion and removal can be integrated in a logarithm time according to the number of line in the document. Indeed, we simply use the binary search algorithm to find the position in the document corresponding to the line identifier.

Also, integration of a delete operation can safely remove the line from the document model, since the total order between remaining lines is not affected. Moreover, this removing will free a position identifier that can be reused. This mechanism reduces the growing rate of line identifier as shown in section 5.

4.4 Correctness of the approach

To ensure convergence in the CRDT framework, the concurrent operations must commute. If line identifiers are unique, non mutable, and totally ordered, the different sites can apply any series of insert operation in any order and obtain the same result.

Lemma 1. *If causality is preserved, Logoot line identifiers are unique (i.e. there cannot be two different lines with the same identifier on one model).*

Proof. The last element of the line identifier contains the unique site number and a local clock of the site which generates the line . Since, the pair (sid, clock) is unique, Logoot line identifiers are unique. □

Theorem 2. *If causality is preserved, Logoot ensure consistency.*

Proof. Since Logoot line identifier are unique, non mutable and totally ordered, every couple of concurrent operation commutes. Thus, Logoot data type is a CRDT. □

To ensure causality, we can use a scalable causal broadcast such as a probabilistic causal broadcast [6]. The idea of such a broadcast is to use causal barriers [14], which size is lower than vector clocks.

Thus, both causality and convergence criteria are respected. The last CCI criterion, i.e. intention, is ensured by the position generation (Definition 3).

5 Evaluation

Theoretically, the size of Logoot line identifiers is unbounded. However, in approaches with tombstones the size of the document model is also unbounded. In the worst case, the Logoot approach has a space overhead superior to the Woot and Treedoc approaches. Indeed, even with the randomization of Logoot, position identifier can grows each time a line is inserted. Thus, if no line is ever removed, the maximum size of the document model overhead is $\sum^n i = O(n^2)$ where n is the total number of line. While Woot and Treedoc approaches have an overhead constant for each line (i.e. $O(n)$).

Theoretically, in the worst case, the Logoot approach has a space line is inserted, even if such cases barely never happen. Thus, if overhead is $\sum^n i = O(n^2)$ where n is the total number of lines. While Woot and Treedoc approaches have a fixed overhead for each In practise, wiki pages contain very few lines (450 words in average) comparing to the number of tombstone while the size of each position identifier remains low. Our approach is sightly less efficient only in one specific case (see section 5.2.3).

In order to effectively measure the Logoot overhead, we have replayed the modifications made on some Wikipedia pages in a Logoot document.

5.1 Methodology

In our implementation, we use 8-bytes integers, hence, a unique identifier contains 16 bytes (one integer and the site identifier).

To replay Wikipedia pages history, we use the Wikimedia API[2] to obtain an XML file containing several revisions of a specific Wikipedia page. Then, using a diff algo-

[2] `http://www.mediawiki.org/wiki/API:Query_-_Properties#revisions_.2F_rv`

rithm [15], we compute the modifications performed between two revisions. Modifications are simply re-executed in our model. Since our approach generates each positions randomly, we re-executed ten times each page history to obtain an average value.

We also measure the overhead for Woot and TreeDoc. The result obtained for TreeDoc does not take account of the “stabledel” and “gc” which aim to remove tombstones. We motivate this choice by the fact that these procedures require to known the exact number of sites which is unknown, unbounded and unstable in a P2P network.

The overhead of Woot and TreeDoc is directly computed from the number and the type of operations performed on the document. Indeed, their overhead is directly proportional to the number of inserted lines in the document since deleted lines remain as tombstones.

We have applied this schema on the top pages of three categories[3][4] :

- The most edited encyclopedic pages.
- The most edited pages.
- The biggest pages.

For each of the treated pages, we present the average – over the last 100 editions – overhead of the Logoot, Woot and Treedoc approaches. We present the average size of the page and the number of patches (i.e. editions on the page).

5.2 Results

Figure 1 (resp. figure 2) shows the relative[5] (resp. absolute) overhead of different approaches on the most edited encyclopedic page of the English Wikipedia. The Logoot’s overhead remains constant all along the editing session, while tombstones based approaches’ overhead continuously grows.

Finally, the Logoot’s overhead is inferior to the document size while tombstones based approaches requires more than 100 times the document size and continuously grows.

5.2.1 Most edited encyclopedic Pages

For instance, while the first page contains only about 553 lines, the number of deletions is about 1.6 millions. As a consequence, tombstone based systems are not well-suited for such documents since we obtain 1.6 millions tombstones for only 553 lines.

Most of the modification done on Wikipedia pages consists in updating the content of some existing lines. Distributed editing systems handle such an update as deleting the old content and inserting the new content. Thus, tombstones number grows quickly.

Also, the figure 1 shows several peaks which are mainly due to vandalism acts. Indeed in the some of the most edited encyclopedic pages of the Wikipedia, there is a lot of vandalism acts done by users, including erasing the whole content of the page. Every vandalism is reverted by re-introducing the previously erased content or removing malicious content introduced. This process adds each time a lot a tombstones (up to

[3] According to `http://en.Wikipedia.org/wiki/Wikipedia:Most_frequently_edited_pages` and `http://en.Wikipedia.org/wiki/Special:LongPages` on end of November 2008.

[4] Due to some technical issues (i.e. invalid characters, missing patch, ...), we skipped some of the top pages, but the first page of each category is presented.

[5] Size of the overhead divided by the size of the visible document on a logarithmic scale.

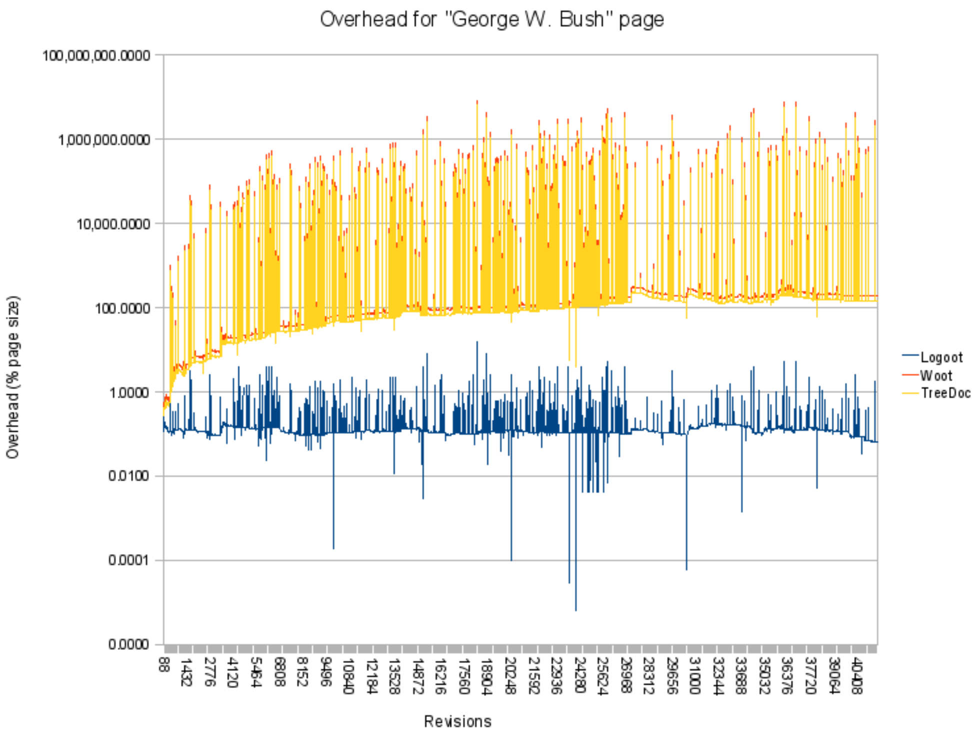

Figure 1: Relative overhead for "George W. Bush" page.

size of the page). Introducing a specific undo mechanism that reuses tombstones such as [13] should reduce the overhead due tombstones.

	Pages	Logoot percent	Wooto percent	TreeDoc percent	Patchs	Size
1	George W. Bush	8.33	16128.75	14590.79	41563	133146
2	List of World Wrestling Entertainment employees	39.24	8413.41	6310.05	27152	16673
3	United States	8.30	5875.07	4406.31	24781	158242
4	Jesus	9.83	4179.09	3134.32	20271	125669
5	2006 Lebanon War	13.62	927.12	695.34	17780	139458
6	Islam	15.92	2996.30	2247.22	15315	101278
7	Roman Catholic Church	5.92	1129.51	847.13	14378	170380
8	Deaths in 2006	18.51	1747.24	1310.43	14029	21880
9	Canada	17.88	4431.19	3323.39	13992	112589
10	Akatsuki (Naruto)	9.81	389.89	292.42	13929	60638
	Average	14.74	4621.76	3715.74	20319	106639

5.2.2 Most edited Pages

These page are discussion pages or special pages mostly edited by bots. In such pages, there is no or very few vandalism but a lot of editions.

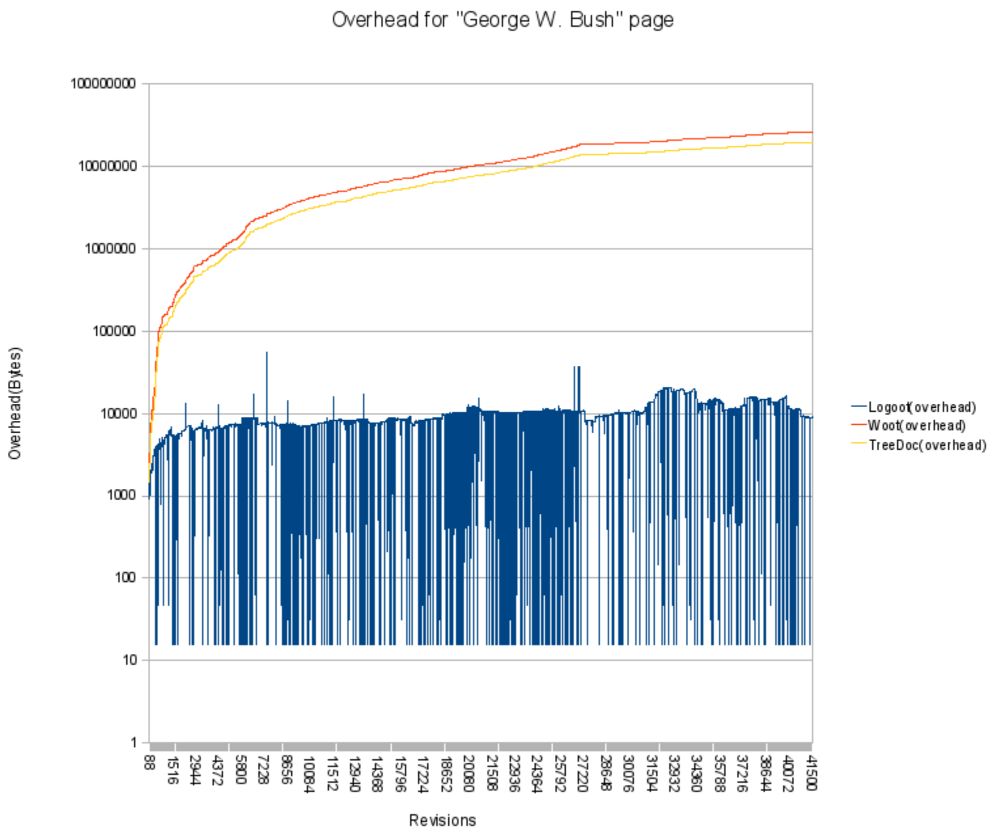

Figure 2: Absolute overhead for "George W. Bush" page.

	Pages	Logoot percent	Wooto percent	TreeDoc percent	Patchs	Size
1	Wikipedia: Administrator intervention against vandalism	27.8	287530.03	215647.52	438330	2369
2	Wikipedia: Reference desk/Miscellaneous	528.28	7492.31	5619.23	148283	133204
3	Wikipedia: Reference desk/Science	189.73	3431.45	2573.59	142722	190858
4	Wikipedia: Introduction	43.74	4195621.30	3146715.98	132693	317
5	Wikipedia: Help desk	59.14	9266.41	6949.81	126509	96256
	Average	169.74	900668.3	675501.23	197707	1011.98

For all these, the Logoot approach is more efficient than tombstone approaches. However, we can notice that the difference is far more important for pages were data are very volatile for instance like case 1 (a communication channel to detected and block vandals) or like case 4 (a sandbox). The over cases represent discussion pages. Users ask questions, and other users reply by modifying the page. Each topic is removed after one week. They are edited in the same way : (mostly) adding content at the end of the page and removing content at the beginning (week old topics). Thus, there is, in these pages, a lot of tombstones but the Logoot identifier grows also.

5.2.3 Biggest Pages

	Pages	Logoot percent	Wooto percent	TreeDoc percent	Patchs	Size
1	Line of succession to the British throne	23.65	488.30	366.23	3317	376760
2	United States at the 2008 Summer Olympics	52.65	314.71	236.03	2314	314748
3	List of sportspeople by nickname	19.14	82.34	61.75	2332	309576
4	List of Brazilian football transfers 2008	27.08	11.33	8.5	752	287128
5	List of college athletic programs by U.S. State	34.60	48.56	36.42	868	305294
6	List of Chinese inventions	5.11	37.71	28.29	2344	293228
7	List of suicide bombings in Iraq since 2003	13.51	24.55	18.42	1260	215763
8	China at the 2008 Summer Olympics	61.55	134.15	100.61	1552	268720
9	List of urban areas in Sweden	40.04	39.61	29.71	19	108353
10	Table of United States Core Based Statistical Areas	63.55	61.54	46.15	31	252236
	Average	34.09	124.28	93.21	1478.9	320899

These pages are often lists of elements. If these lists are always edited in the same way (for instance adding elements at the end of the page), they represent the worst cases for our approach. Indeed, the Logoot line identifier will grow the quickest, especially if insertions are done in many different occasions. Effectively in case 4, our approach is less efficient than tombstone approaches.

This behaviour can be improved by changing the choice of the integer position during insertion. Instead of a random choice, a position between p and q, can be set at $p + 1$ (or at $p + k$). When a site detects that the line identifiers grows too quickly on a page, it can change its strategy. Since any positions between p and q can be choose, a site can independently change its strategy without affecting or even informing the others. Such a modification should also improves performance for "help desk" cases of most edited pages.

However, these results show our approach is in average, even in these disadvantageous real cases, less costly than tombstone approaches.

5.3 Limits of the experimentation

Since the Wikipedia uses a centralized wiki, we can expect a slightly different behavior in a P2P system.

- Thanks to its centralized architecture, the Wikipedia reduces the impact of concurrent modifications. Assume that two users are editing the same wiki page at the same time. The first user saves its modifications. When the second user wants to save, the Wikipedia notifies him that a concurrent version was produced. Therefore, the second user modifies the page to integrate both modifications. Finally, the second user's modifications is not concurrent to the first user's modification.

  In a P2P environment, preventing users to make concurrent modifications is not a realistic hypothesis. Therefore, concurrent modifications are automatically merged. This will certainly produce an "inconsistent" document which requires the intervention of some user to correct it. Therefore, the number of editions will certainly be more important in a P2P wiki than in centralized wiki.

- In Wikipedia, some pages are protected to reduce the number of vandalism acts. However, such protection mechanism is not compatible with P2P constraints.

Therefore, in a P2P wiki, the number of vandalism acts is certainly more important. Therefore, we expect to obtain more edits and vandalism acts on a P2P wiki system.

- Contrary to the Woot approach, CRDT approaches, including ours, requires a causal broadcast to achieve convergence. However, a causal delivery implies an overhead on each message sent by each site. The two main mechanism to achieve a causal delivery are vector clocks [16] and causal barriers [14]. Vector clocks are not usable in P2P networks since their sizes are proportional to the number of site. Causal barriers have a smaller size, that depend only on the degree of concurrency of the operation in the network. On collaborative editing system, this degree remains low : less than 3 edits per second on the whole English Wikipedia in average[6]. However, a realistic measure of the communication overhead can only be achieved with a corpus of concurrent collaborative editions.

6 Conclusion

In this report, we have presented Logoot, an approach to build a P2P collaborative editor. Compared to existing approaches, the main advantage of Logoot is that it does not require tombstones. Thus the space overhead remains constant during the life of the document, and the performances do not decrease continually. Also, it does not require a tombstone garbage collecting mechanism which is very costly (consensus) or even unusable in practice in P2P networks.

The experimentation shows that Logoot's unbounded list of identifier given to each line stay small in practice and is much more suitable than tombstones.

An other contribution of this paper is to provide a corpus of experiment to test collaborative editor on realistic conditions. An interesting fact about Wikipedia is that wiki is not only used to edit the encyclopedic pages but also to manage all the process around the Wikipedia (vandals detection, discussions, page status votes, ...). Our approach is particularly suited for this kind of page where a lot of edit are made and data are not persistent.

References

[1] G. Oster, P. Urso, P. Molli, and A. Imine, "Tombstone transformation functions for ensuring consistency in collaborative editing systems," in *The Second International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2006)*. Atlanta, Georgia, USA: IEEE Press, November 2006.

[2] S. Weiss, P. Urso, and P. Molli, "Wooki: a p2p wiki-based collaborative writing tool," in *Web Information Systems Engineering*. Nancy, France: Springer, December 2007.

[3] M. Shapiro and N. Preguia, "Designing a commutative replicated data type," INRIA, Rapport de recherche INRIA RR-6320, October 2007. [Online]. Available: http://hal.inria.fr/inria-00177693/fr/

[6] 2.69 in October 2008 according to `http://en.wikipedia.org/wiki/Wikipedia:Statistics`

[4] H.-G. Roh, J. Kim, and J. Lee, "How to design optimistic operations for peer-to-peer replication," in *JCIS*, 2006.

[5] Y. Saito and M. Shapiro, "Optimistic replication," *ACM Computing Surveys*, vol. 37, no. 1, pp. 42–81, 2005.

[6] P. T. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec, "Lightweight probabilistic broadcast," *ACM Trans. Comput. Syst.*, vol. 21, no. 4, pp. 341–374, 2003.

[7] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenkcr, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," in *Proceedings of the sixteenth annual ACM Symposium on Principles of Distributed Computing - PODC'87*. ACM Press, 1987, pp. 1–12.

[8] C. Sun, X. Jia, Y. Zhang, Y. Yang, and D. Chen, "Achieving convergence, causality preservation, and intention preservation in real-time cooperative editing systems," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 5, no. 1, pp. 63–108, Mars 1998.

[9] L. Lamport, "Time, clocks, and the ordering of events in a distributed system." *Commun. ACM*, vol. 21, no. 7, pp. 558–565, 1978.

[10] G. Oster, P. Urso, P. Molli, and A. Imine, "Data Consistency for P2P Collaborative Editing," in *Proceedings of the ACM Conference on Computer-Supported Cooperative Work - CSCW 2006*. Banff, Alberta, Canada: ACM Press, nov 2006, pp. 259–267.

[11] M. Cart and J. Ferri, "Asynchronous reconciliation based on operational transformation for p2p collaborative environments," in *CollaborateCom*, 2007.

[12] C. A. Ellis and S. J. Gibbs, "Concurrency control in groupware systems." in *SIGMOD Conference*, J. Clifford, B. G. Lindsay, and D. Maier, Eds. ACM Press, 1989, pp. 399–407.

[13] S. Weiss, P. Urso, and P. Molli, "An undo framework for p2p collaborative editing," in *CollaborateCom*, Orlando, USA, November 2008.

[14] R. Prakash, M. Raynal, and M. Singhal, "An adaptive causal ordering algorithm suited to mobile computing environments," *J. Parallel Distrib. Comput.*, vol. 41, no. 2, pp. 190–204, 1997.

[15] E. W. Myers, "An o(nd) difference algorithm and its variations," *Algorithmica*, vol. 1, no. 2, pp. 251–266, 1986.

[16] F. Mattern, "Virtual time and global states of distributed systems," in *Proceedings of the International Workshop on Parallel and Distributed Algorithms*, M. C. et al., Ed. Chteau de Bonas, France: Elsevier Science Publishers, Octobre 1989, pp. 215–226.

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399