Performance of real-time collaborative editors at large scale: User perspective

Quang-Vinh Dang, Claudia-Lavinia Ignat

Performance of real-time collaborative editors at large scale: user perspective

Quang-Vinh Dang
Université de Lorraine, LORIA, F-54506
Inria, F-54600
CNRS, LORIA, F-54506
quang-vinh.dang@inria.fr

Claudia-Lavinia Ignat
Inria, F-54600
Université de Lorraine, LORIA, F-54506
CNRS, LORIA, F-54506
claudia.ignat@inria.fr

***Abstract*—Real-time collaborative editing allows multiple users to edit a shared document at the same time. It received a lot of attention from both industry and academia and gained in popularity due to the wide availability of free services such as Google Docs. While these collaborative editing systems were initially used in scenarios involving only a small set of users such as for writing a research article, nowadays we notice a change in the scale from several users to communities of users. Group note taking during lectures or conferences is an emerging practice.**

An important measure of performance of real-time collaborative editing systems is delay. Delays exist between the execution of one user modification and the visibility of this modification to the other users. They can be caused by network physical communication media, complexity of consistency maintenance algorithms and system architecture. Some user studies have shown that delay affects group performance in collaborative editing. In this paper, we measure delays in popular real-time collaborative editing systems such as Google Docs and Etherpad and we study whether these systems could cope with large scale settings from a user perspective. Our results show that these systems are not yet ready for large-scale collaborative activities as either they reject new users connection or a high delay appears when facing an increasing number of users or their typing speeds in the same shared document.

I. Introduction

Today team working is a key role of success in companies or organizations. Very often members within an organisation or between different collaborating organisations are located at different geographical places and can work at different times. For an effective collaboration, team members usually need to use collaborative tools in order to overcome the geographical distance. Real-time collaborative editing systems are commonly used as they allow multiple users to edit a shared document at the same time.

Benefits of real time on-line collaborative editors are multiple. Firstly, they provide a ready-to-use platform for all users to view and modify documents on their web browsers, without installing heavy software bundle such as Microsoft Office or Libre Office. Secondly, they provide an environment where multiple users can contribute to shared documents in a fast and easy manner. While sharing documents by emails or physical mediums such as USB sticks would require to manually deal with multiple concurrent revisions and using version control systems such as git and svn would require trained users, in real-time collaborative editing merging is automatically performed without any user intervention.

Business analysis showed that new cloud collaborative editing systems such as Google Drive are taking the market share from the traditional document software provider such as Microsoft Office [1]. The number of users of Google Drive service increased from 10 millions on 2012 [2] to 240 millions on October 2014 [3]. On September 2015 Google Drive announced that there are one million paying customers using their service [4].

Initially, real-time collaborative editing systems were used in scenarios involving a small number of concurrent editing users (e.g, up to ten) such as writing a research article or a brainstorming session. However, scenarios involving a large number of concurrent editing users are emerging, such as students of a class or participants in a conference that collaboratively take notes. A recent example is a MOOC (Massive Open Online Course) where the 40,000 participants were asked to access to parts of the Google Docs documents created for the course. Due to the high number of concurrent edits to the same documents, the system crashed and finally the lecture was cancelled [5].

Various quality aspects should be taken into consideration in the design phase of large-scale collaborative editing systems. One of the important requirements of these systems is delay [6], [7]. Delays exist between the execution of one user's modification and the visibility of this modification to the other users. Delays can be caused by different reasons: network delay due to physical communication technology be it copper wire, optical fiber or radio transmission; complexity of various algorithms for ensuring consistency, where most of them depend on the number of users and number of operations that users performed; the type of architectures: For thin client architectures the computation for algorithms for maintaining consistency is done mainly on the server, which becomes a bottleneck in the case of a large number of users and operations, and therefore causes an increased delay for seeing operations of other users. For thick client architectures the computation is done mainly on the client side and delays are lower in this case.

In the context of collaborative editing, the delay is a critical

ISBN 978-3-901882-83-8 © 2016 IFIP

concern and it has a great influence on the performance and behavior of users [7]. By studying users' behavior in real-time collaborative note taking with artificial added delay, Ignat et al [6] claimed that "delay increases grammatical errors and redundancy, resulting in a decreased quality of the task content". However, delay has not been addressed at its desired level in the development of real-time collaborative editing systems.

In this paper, we aim at measuring the performance of real-time collaborative editing systems from users point of view. More precisely, we measure the perceived delay by users in online real-time collaborative editing systems in their normal working environment, i.e. using web browsers.

Setting up an experiment with numerous real users that edit concurrently a shared document would not be possible with current tools. Existing tools restrict the number of users editing a document and most of them are not open-source in order to allow code instrumentation for delay measurement. We instead simulated user behavior by means of agents that use popular web-based real-time collaborative editing services currently available in the market: Google Docs[1] and Etherpad[2].

The paper is structured as follows. We start by describing the set-up of our experiments: how we modeled and simulated user behavior and how we measured delays. We next present the performance evaluation of Google Docs and Etherpad and discuss the results obtained. We finally provide concluding remarks.

II. Evaluation settings

A. User Behavior

We define user behavior as a set of the following actions, which can be further extended:

- Start a web browser instance (Firefox, Chrome, Internet Explorer, Safari, Edge, etc.).
- Surf to the dedicated web page of a collaborative editor.
- Load a shared document.
- Perform modifications by inserting and/or deleting characters in the document.
- Interact with buttons on the web page by using mouse / pointing system.
- Close the web browser, since current collaborative editors automatically save user changes.

B. Simulation Settings

In order to simulate the real user behavior on web browsers, we selected Selenium [8], which has been widely accepted in web-based testing community [9].

The simulation is distributed on multiple computers[3]. Three types of simulated users have been defined:

1) Writer: writes a specific string to the shared document.
2) Reader: waits and reads the specific string from the writer.
3) DummyWriter: writes random strings to the shared document. Random strings are different from the specific string. DummyWriters are used to simulate concurrent users.

Each simulator (Writer, Reader, DummyWriter) performs its task on different Google Chrome browser window. The delay is measured by the time period between the moment the specific string is written by the Writer and the moment when the specific string is read by the Reader. In order to avoid clock synchronisation issues, both Reader and Writer are executed on the same computer.

C. Experiment Settings

For each real-time collaborative editing system, i.e. Google Docs and Etherpad, we measured the performance (delay) in different settings by varying the number of users who modify the document at the same time, and their typing speed, i.e. the number of characters each user types to the document in one second.

As the number of users that can concurrently modify a document in Google Docs[4] is limited to 50, we varied the number of users from one to 50. The usual range of user typing speed is 2–4 characters per second [10]. We also considered that higher speeds could be achieved by performing cut and paste operations on large blocks of text. We therefore varied the typing speed from one to 8 characters per second.

We created five shared documents and then evaluated the delays in turn on each of these documents and for each combination of settings (number of users and typing speed). In order to further eliminate random effects on the performance achieved, for each of the shared document and for each combination of settings, we repeated the experiment four times.

We used five local computers located at Inria Grand-Est, Nancy, France with the corresponding configuration features described in Table I. Clients simulating user behavior were executed on one of these computers: the Writer and Reader are executed by the first computer (with CPU Intel i7 720QM), and the DummyWriters are executed by other computers. DummyWriters are assigned to computers in a load balancing fashion: during the experiment when the number of clients, i.e. DummyWriters, is increased, each new DummyWriter is executed on the computer running the minimum number of clients with respect to its capacity, i.e. CPU and memory. We report the maximum number of DummyWriter which are executed on each computer in the third column of Table I.

III. Results

In this section, we present and discuss the performance evaluation results of two popular real-time collaborative editing systems: Google Docs and Etherpad.

[1]https://docs.google.com

[2]http://etherpad.org

[3]The implementation is available at https://github.com/vinhqdang/collaborative_editing_measurement

[4]https://support.google.com/docs/answer/2494827?hl=en as on 15-Feb-2016.

TABLE I
THE EXPERIMENT CONFIGURATION

CPU	Memory	Number of simulated users
Intel i7 720QM	8 GB	2
Intel Xeon W3550	8 GB	15
Intel Xeon W3520	8 GB	15
Intel Core 2 Duo E6850	4 GB	10
Intel Core 2 Duo E6550	4 GB	9

A. Google Docs

Google Docs is the most popular real-time collaborative editing system today. The service was introduced in 2007, and quickly attracted over one million users [1], [2].

The results of performance evaluation of Google Docs are displayed in Figures 1–5 for different typing speeds. The delays mean value line depicted in each figure shows the increasing trend of delay with an increasing number of users that join and modify the shared documents at the same time.

The above graphs show a very interesting feature of Google Docs. When the number of users is less than ten, Google Docs provides a very good and stable performance. The delays are very small and stable meaning that the performance of the system has not been affected when the number of users increases from one to ten. However, when the number of users exceeds ten, the performance of Google Docs decreases quickly, meaning that the delay increases significantly. This might be an explanation for the limit of 50 concurrent users specified by the Google Docs documentation.

The results also show us another interesting property of Google Docs: a higher typing speed leads to a higher dependency of delay on number of users. In other words, a higher typing speed will lead to a higher delay, and the delay also increases faster with the number of users.

We notice that it is very common to observe delays over ten seconds with Google Docs. Moreover, even if Google Docs documentation claimed that up to 50 users can modify a shared document at the same time, it is not always the case. We only can simulate 50 users if the typing speed of users is one character per second. Otherwise, if we increase the typing speed, a maximum of 38 users can log in and use the service. Additional users cannot use the system as they are repeatedly displayed the following message at login “Wow this file is really popular! Some tools might be unavailable until the crowd clears”.

B. Etherpad

Etherpad is a popular open-source web based collaborative platform, with the first version being released in 2008. Etherpad is currently being used by many open-source and non-profit organizations, such as Wikimedia[5].

In order to evaluate the performance of Etherpad, we installed the source code provided by Etherpad development team[6] on our own server (Intel Xeon W3550) and performed

[5]https://etherpad.wikimedia.org
[6]http://etherpad.org/\#download

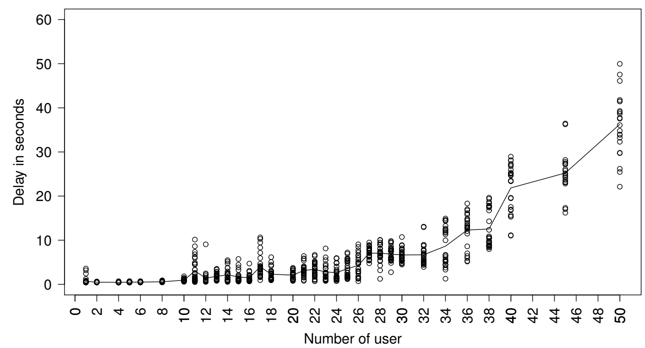

Fig. 1. Performance of Google Docs with a typing speed of one character / second

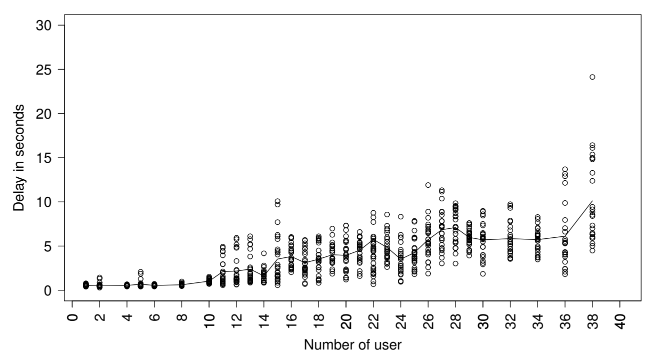

Fig. 2. Performance of Google Docs with a typing speed of two characters / second

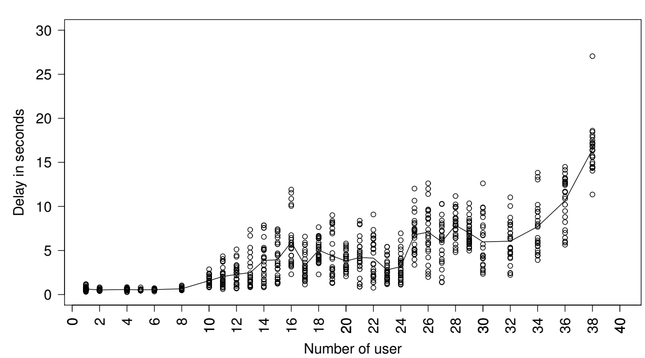

Fig. 3. Performance of Google Docs with a typing speed of four characters / second

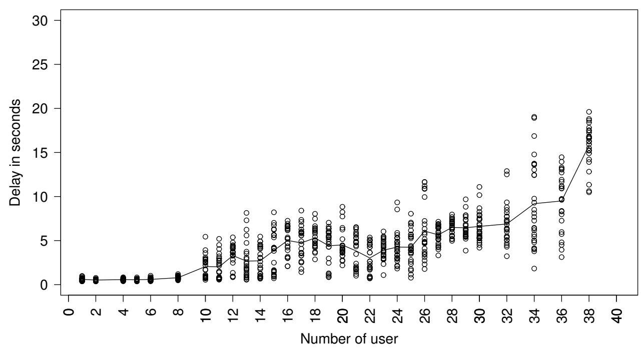

Fig. 4. Performance of Google Docs with a typing speed of six characters / second

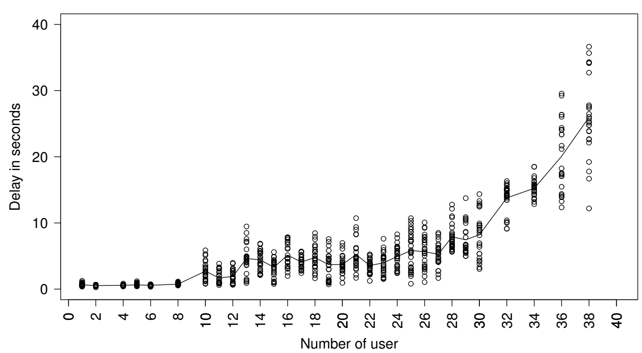

Fig. 5. Performance of Google Docs with a typing speed of eight characters / second

the same evaluation method that we previously described for Google Docs. We applied all default settings of Etherpad (i.e. we used dirtyDB as the underlying database), and maintained the same evaluation settings as in the case of Google Docs.

When the number of users is less than ten, we observe a similar phenomenon as Google Docs: Etherpad responds quickly to users' modifications, and delays are small and stable. However, when the number of users exceeds ten, Etherpad starts rejecting connections from new users, as it can be seen in the screenshot in Figure 6.

We can conclude that, Etherpad cannot be used by more than ten users that concurrently modify the shared document at the same time.

IV. Discussion

A. Source of delay

As previously mentioned, in real-time collaborative editing, delays can appear because of network due to the structure and configuration of the Internet, which basically operates on the "best effort" principle, trying to deliver data from a computer to other ones as fast as possible but without any guarantee of time bound. Delays are also due to the collaborative editing systems architecture such as client-server, peer-to-peer, thin or

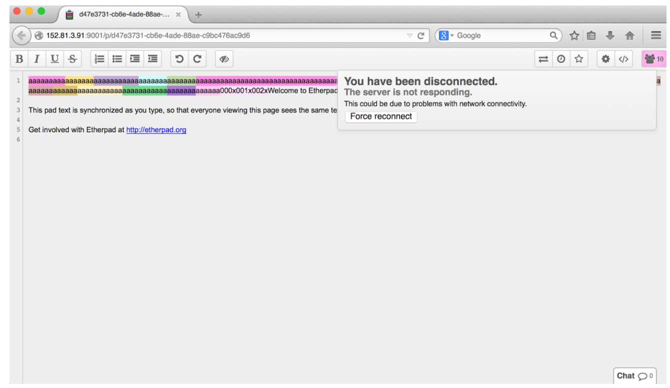

Fig. 6. Etherpad rejects users' connection when the number of users exceeds ten

thick client. Moreover, delays are due to the synchronisation algorithms implemented by each system.

In this paper, we measured the delay with different number of users and typing speeds. The delay the real users observe depends on how many users are modifying the sharing document at the same time, and how fast they are typing. However, we could not perform a more in-depth analysis on the reasons of delay as information about the inside architecture and algorithms used in Google Docs is not available and we could not take into account all the network infrastructure from our computers to Google Docs server.

During the experiments, we monitored the network bandwidth and CPU usage on our experimental computers[7]. The monitoring logs showed that our computers never consume more than 30% of CPU usage, and the incoming/outgoing network bandwidth is always less than 100 Kilobytes per second, which can be easily satisfied with the Internet connection today [11].

B. Sharing limit

Both Google Docs and Etherpad, the two popular real-time collaborative editing systems today, do not support a large number of users. Etherpad stops to accept new incoming users when the number of current users is more than ten. Google Docs claims that the service can support up to 50 users that can modify the shared document at the same time. However, this is true only if these users type at slow speed (1 character / second). Otherwise, the service cannot support more than 38 users that modify the shared document at the same time.

Therefore, Google Docs and Etherpad are not suitable for large-scale collaborative editing activities today, where hundreds of persons can share and modify a document such as a notes in events, meetings, conferences or MOOCs.

C. Effect on users

The effect of delay on users' behavior in collaborative editing has been studied in [6], [12]. Twenty groups of four students from a French university have been recruited to

[7]CPU usage and network bandwidth were monitored by using *multiload indicator*, which is available at https://launchpad.net/indicator-multiload

perform an experiment including several tasks in collaborative editing. Users were asked to perform each collaborative task using the Etherpad collaborative editor under instructions that demanded interleaved work. Users were allowed to coordinate themselves by using the chat available in Etherpad. Without informing participants, delays were intentionally added, i.e. the server has been programmed to wait a certain amount of time before sending data to clients. The artificial delay levels included two, four, six, eight and ten seconds, which are realistic as we discussed in Section III. Each group had to perform the required tasks under a constant but undeclared delay in the propagation of changes between group members. Software recorded each user's desktop activity, including task performance as well as chat for coordination.

The task where users were provided with a list of movies and asked to search for the release date of those movies and then sort them in an ascending order according to the release date was analysed in [12]. For the analysis of the collected data outcome metrics for measuring the quality of the realised task but also process metrics for analysing user behaviour during achievement of the tasks were introduced: a sorting accuracy based on the insertion sort algorithm, average time per entry, strategies (tightly coupled or loosely coupled task decomposition of the task), chat behavior and collisions between users. It has been found that delay slows down participants which decrements the outcome metric of sorting accuracy. Tightly coupled task decomposition enhances outcome at minimal delay, but participants slow down with higher delays. A loosely coupled task decomposition at the beginning leaves a poorly coordinated tightly coupled sorting at the end, requiring more coordination as delay increases.

The note taking task where users listened to an interview about cloud computing and took notes during this time was analysed in [6]. It has been noticed that due to delay, notes about the same topic were taken two, three and even four times. What happened is that when two users want to take notes on the same topic, in the presence of delay, changes of one user are not immediately visible to the other user, so a user thinks that the other user is not taking notes, so he/she is taking the notes. In that way, finally, the notes are in double. If more than two users try to take notes simultaneously, finally the same idea will appear three or even four times. It was found that the error rate is higher for groups that experienced a higher level of delay and redundancy is higher for groups in higher delay condition. Moreover, as delay increases the keywords depicted by users decreases. Groups were classified into two categories according to their declared experience in the domain of collaborative editing. For high experienced groups redundancy increases with the delay, but for low experienced groups the same tendency could not be observed. Chat behavior by means of number of accord words and definite determiners which together provides a common ground knowledge was considered as a measure of coordination. Low experienced groups used more coordination to manage redundancy. High experienced groups did not adjust their collaboration effort to manage redundancy.

A general hindrance of delay was observed in all analysed tasks. Delay destroys the value of collaborative editing and forces independent, redundant work.

Delays measured in GoogleDocs when the number of users exceeds ten are largely superior to the artificial delays experienced in [6], [12]. We therefore expect that a high hindrance of delay will be experienced in Google Docs in scenarios of collaborative editing that involve a large number of users that concurrently modify a shared document.

D. Implications for design

Our primary purpose is to demonstrate the delay users could observe in popular real-time collaborative editing systems available in the market, and we showed that delay is a fact in current real-time collaborative editing systems. In the scale of more than ten users to modify the sharing document at the same time, delay can come up to 50 seconds.

Usually, in distributed computing, delays could appear due to traffic congestion when the amount of data transferred between nodes overcomes the capability of the network [13]. However, as we presented in Section IV-A, we did not observe the traffic congestion during our experiments. Therefore, we could suggest that the delay mostly comes from the architecture and implementation of the services.

As claimed in the official blog of Google Drive development team [14], Google Docs is relying on the Jupiter algorithm [15] for synchronization between nodes. This might not be the best choice of algorithm to be used in collaborative editing at large scale, because the Jupiter algorithm, which belongs to Operational Transformation family of synchronisation solutions [16] requires a lot of computation on the server side, which increases the delay users observe in large-scale settings. Different algorithms such as CRDT [17], [18], standing for *Conflict-free Replicated Data Type* should achieve a better performance in large-scale collaborative editing and feature smaller delays [19]. Moreover, CRDT-based algorithms on strings should achieve better performance than character-based CRDT algorithms [20], [21].

Notifying delay to users could be implemented in real-time collaborative editing services. As suggested by several researches [6], [7], notified delay could let the users adapt their behaviors for the context. However, the delay notification has not been implemented yet in the real-time collaborative editing systems.

V. Conclusions

In this paper, we presented the performance measurement in term of delay in popular real-time editing services in the market. We demonstrated that, delay is a fact and high delay will appear and increase if there are more users joining to modify the shared document at the same time, or the users increase their typing speed. We demonstrated that the existing real-time collaborative systems are not yet ready for large-scale collaborative activities, as they reject the new users' connection if the number of users in the system increases a certain limit.

Delay destroys the value of collaborative editing and forces independent, redundant work. Therefore, delay should be avoided and needs more attention from development team. We discussed several strategies to tackle the problem such as using suitable consistency maintenance algorithms or notifying delay to users.

VI. Acknowledgements

This work has been carried out thanks to the support of the PSPC OpenPaaS::NG project funded by the "Investissements d'Avenir" French government program managed by the "Commissariat général à l'investissement" (CGI).

References

[1] A. Covert, "Will google docs kill off microsoft office?" 2013. [Online]. Available: http://money.cnn.com/2013/11/13/technology/enterprise/microsoft-office-google-docs/

[2] J. Crook, "Google drive now has 10 million users: Available on ios and chrome os," 2012. [Online]. Available: http://techcrunch.com/2012/06/28/google-drive-now-has-10-million-users-available-on-ios-and-chrome-os-offline-editing-in-docs/

[3] E. Protalinski, "Google announces 10% price cut for all compute engine instances, google drive has passed 240m active users," 2014. [Online]. Available: http://thenextweb.com/google/2014/10/01/google-announces-10-price-cut-compute-engine-instances-google-drive-passed-240m-active-users/

[4] B. Darrow, "Google drive claims one million paying customers, er, organizations," 2015. [Online]. Available: http://fortune.com/2015/09/21/google-drive-1m-paid-users/

[5] S. Jaschik, "Mooc mess," 2013. [Online]. Available: https://www.insidehighered.com/news/2013/02/04/coursera-forced-call-mooc-amid-complaints-about-course

[6] C.-L. Ignat, G. Oster, O. Fox, V. L. Shalin, and F. Charoy, "How do user groups cope with delay in real-time collaborative note taking," in *Proceedings of the 14th European Conference on Computer Supported Cooperative Work (ECSCW)*, 2015, pp. 223–242.

[7] I. Vaghi, C. Greenhalgh, and S. Benford, "Coping with inconsistency due to network delays in collaborative virtual environments," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST)*, 1999, pp. 42–49.

[8] A. Holmes and M. Kellogg, "Automating functional tests using selenium," in *Proceedings of AGILE Conference (AGILE)*, 2006, pp. 270–275.

[9] R. Angmo and M. Sharma, "Performance evaluation of web based automation testing tools," in *Proceedings of the 5th International Conference Confluence, The Next Generation Information Technology Summit (Confluence)*, 2014, pp. 731–735.

[10] R. William Soukoreff and I. Scott Mackenzie, "Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard," *Behaviour & Information Technology*, vol. 14, no. 6, pp. 370–379, 1995.

[11] Akamai, "State of the internet report," 2015. [Online]. Available: https://www.akamai.com/us/en/our-thinking/state-of-the-internet-report/

[12] C.-L. Ignat, G. Oster, M. Newman, V. Shalin, and F. Charoy, "Studying the effect of delay on group performance in collaborative editing," in *Proceedings of the 6th International Conference on Cooperative Design, Visualization and Engineering (CDVE)*, 2014, pp. 191 – 198.

[13] J. F. Kurose and K. W. Ross, *Computer networking: a top-down approach*. Addison-Wesley, 2007.

[14] G. D. Blog, "What's different about the new google docs: Making collaboration fast," 2010. [Online]. Available: http://googledrive.blogspot.com/2010/09/whats-different-about-new-google-docs.html

[15] D. A. Nichols, P. Curtis, M. Dixon, and J. Lamping, "High-latency, Low-bandwidth Windowing in the Jupiter Collaboration System," in *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology (UIST)*, 1995, pp. 111–120.

[16] C. A. Ellis and S. J. Gibbs, "Concurrency control in groupware systems," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 1989, pp. 399–407.

[17] S. Weiss, P. Urso, and P. Molli, "Logoot : A Scalable Optimistic Replication Algorithm for Collaborative Editing on P2P Networks," in *Proceedings of the 29th International Conference on Distributed Computing Systems (ICDCS)*, 2009, pp. 404–412.

[18] N. Preguiça, J. M. Marquès, M. Shapiro, and M. Letia, "A Commutative Replicated Data Type for Cooperative Editing," in *Proceedings of the 29th International Conference on Distributed Computing Systems (ICDCS)*, 2009, pp. 395–403.

[19] M. Ahmed-Nacer, C. Ignat, G. Oster, H. Roh, and P. Urso, "Evaluating crdts for real-time document editing," in *Proceedings of the ACM Symposium on Document Engineering (DocEng)*, 2011, pp. 103–112.

[20] L. André, S. Martin, G. Oster, and C.-L. Ignat, "Supporting Adaptable Granularity of Changes for Massive-scale Collaborative Editing," in *Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2013.

[21] W. Yu, L. André, and C.-L. Ignat, "A CRDT Supporting Selective Undo for Collaborative Text Editing," in *Proceedings of the 10th International Federated Conference on Distributed Computing Techniques (DisCoTec) Distributed Applications and Interoperable Systems (DAIS)*, 2015, pp. 193–206.