Decentralized Creation of Academic Documents Using a Network Attached Storage (NAS) Server

Johannes Wilm[1], Afshin Sadeghi[2], Christoph Lange[3], Philipp Mayr[4].

[1] GESIS–Leibniz Institute for the Social Sciences, Unter Sachsenhausen 6-8, 50667 Köln, Germany, mail@johanneswilm.org
[2] University of Bonn, Römerstraße 164, 53117 Bonn, Germany, sadeghi@cs.uni-bonn.de
[3] University of Bonn, Römerstraße 164, 53117 Bonn, Germany and Fraunhofer IAIS, Schloss Birlinghoven, 53754 Sankt Augustin, Germany, langec@cs.uni-bonn.de
[4] GESIS–Leibniz Institute for the Social Sciences, Unter Sachsenhausen 6-8, 50667 Köln, Germany, philipp.mayr@gesis.org

Event: Linked Research workshop at ESWC 2017

Identifier: https://www.fiduswriter.org/2017/05/24/decentralized-creation-of-academic-documents-using-a-network-attached-storage-nas-server/

In reply to: https://linkedresearch.org/calls

Published: 2017-06-13

Abstract. Scholarly document creation continues to face various obstacles. Scholarly text production requires more complex word processors than other forms of texts because of the complex structures of citations, formulas and figures. The perceived need for peer review, often single-blind or double-blind, creates needs for document management that other texts do not require. Additionally, the need for collaborative editing, security and strict document access rules means that many existing word processors are imperfect solutions for academics. Nevertheless, most papers continue to be written using Microsoft Word. In this position paper we analyze some of the problems with existing academic solutions and then present an argument why we believe that running an open source academic writing solution for academic purposes, such as Fidus Writer, on a Network Attached Storage (NAS) server could be a viable alternative.

Keywords: Network Attached Storage server, Linux, editing, collaborative editing.

1 Issues with existing approaches

1.1 Issues with Microsoft Word and other general word processors for academics

Microsoft Word has the advantage that users are familiar with its easy to use What-You-See-Is-What-You-Get (WYSIWYG) interface. However, articles authored in Word have the problem that they lack semantic information, which means that conversion for final publication into other formats will be more difficult and will require human intervention. Because the conversion process is imperfect, there is also the chance of loss of information or misinterpretation on the part of the human executing the conversation. The same is true for open source alternatives with a similar user interface and workflow, such as LibreOffice or OpenOffice.org.

Additional problems occur if one needs to collaborate on a text among several writers: Collaborative editors such as Google Docs or Microsoft Office 365 Online place the documents on servers outside of the control of the user, and potentially confidential information is shared with companies operating servers that could be targeted by espionage. Espionage may be even easier if information is stored in a standard data store companies.

One may argue that information that is to be published will be available anyway. This is true for a lot of fields, but there are some fields where even published material is only made accessible to a small amount of people. And even in those circles where it is made available publicly at the time of publication, not everyone is comfortable with giving specific parties early access to the material.

1.2 Issues with web services targeting academic writers

Some new online writing services have appeared in recent years[1] to target academic writers specifically. These editors handle citations, formulas and the like, and conversions to final output formats will therefore require less or no human intervention.

While these editors may have solved much of the conversion issues, in so far as they are closed source applications hosted by a single company, they have the same security issues as Google Docs and Microsoft Office 365 Online. Open source

1 Among the academic text editing apps that have appeared are: Authorea, Fidus Writer, ShareLaTeX and WriteLaTeX/Overleaf. Other, decentralized, editors exist, such as Dokieli or Laverna, but due to the way they are opeting, they are not able to provide realtime collaboration.

alternatives, such as Fidus Writer and ShareLaTeX, in theory have less of this issue, as the application can be installed on a secure server. In practice, most users will not have their own regular server, so that this option is not really accessible to them. Should they hire server space from a third party, they may have more control over which country their data will be stored in, but they will still be exposing their data to the company operating the server.

1.3 Issues with decentralized document editing in a collaborative way

An alternative for decentralized collaboration is one where the editing application is installed on the end user's computer. This solves the problem of the server, but it creates a number of other challenges:

2 General installation processes are too complex for novice users. The fact that the developers of the editing software cannot know which OS their end users will be running makes it hard to give standardized installation instructions.

3 Users today are often working on sections of the Internet behind routers with changing IP numbers on the internet. Permanent IP numbers are given out by Internet Service Providers (ISPs) only for an extra fee or not provided at all. Without any other aids, it is therefore somewhat tricky to connect two computers with each other if they are on different local networks. While there are ways around this problem accessible to IT professionals, it will likely be too complex for the average end user.

4 If two users are collaborating on writing a document, but they are working at different times and cannot guarantee that either computer is on the Internet the entire time, merging changes becomes a problem. Even though merging mechanisms may find a way to automatically merge changes. There is no guarantee that the human language and argument described in the text still make sense if texts are not merged immediately and writers can be guaranteed that the version they are working on contains all the additions made by other collaborators.

Taken together, these points mean that a direct real-time collaboration setup cannot reasonably be established and run by computer novices without significant help from IT professionals.

5 Decentralized document editing using a Network Attached Storage (NAS) Server

Another option is the installation of a small and local server on the side of one of the document editors. Even search giant Google, known for its various cloud services, recognizes the need for data stored locally on smaller servers as it offers its Google Search Appliance product for companies with large amounts of data[2]. NAS servers can fulfill a similar role, but scaled to the comparatively smaller amount of data needed within an academic text editing setting.

NAS servers solve some of the issues encountered when trying to run collaboration from the end user's computer: If one targets one specific NAS platform, the installation process can be simplified to a few clicks and filling in forms that even novice computer users can handle. The lack of a permanent IP address is made up by a dynamic DNS service offered by the vendor of the NAS server. As the installation and management procedures of NAS servers generally take place through web interfaces, they are made to be capable of serving at least basic web pages. As long as the NAS Server can be turned on and connected to the Internet constantly, it does not matter that the end users connect to the document at different times. Another alternative may be a mini-computer such as inexpensive Linux computers that are sold within a similar price range of around 90-200 Euros. Different from NAS servers, they are targeting more advanced users and do not always come with the same dynamic DNS service built-in.

The usage of NAS servers for this purpose creates some other challenges however, as the main purpose of the devices is that of a storage device for backup of files and not a general web server for real-time collaboration. CPU power and RAM are therefore somewhat limited. Also, the NAS represents an extra cost to the end user, which means that the more costly devices will not be an option in many cases.

6 Test setup

In order to find out whether a NAS Server would be a practical alternative, we took a NAS device from the lower end of the spectrum – a Synology DS215J that had been running online for 2.5 years for backup purposes – and we tried to install Fidus Writer on it. Fidus Writer was chosen because it targets specifically computer novices in the humanities and social sciences who require a WYSIWYG user interface that still provides all the features needed for scientific text editing [1]. Other features that are particularly useful are citation and figure management, formula editing as well as

2 https://enterprise.google.com/search/products/gsa.html

export into a number of formats commonly used by journals, such as LaTeX and Open Document Text (ODT). It has also recently gained the ability to be combined with the peer-review management system Open Journal Systems (OJS) [2].

The DS215J has a 800 MhZ Marvell Armada 375 Dual Core CPU and 512 MB of RAM. Currently the successor version of the DS215J, the DS216J, sells for around 163 Euros (May 2017)[3]. The NAS was installed on a home network connected to the internet with a 51.4 MBit (Down)/10 MBit (Up) Deutsche Telekom connection, in northern Germany. Tests were performed from southern Sweden with a 55.5 MBit (Down)/47 MBit (up) Telecom 3 Sverige AB connection. Total air distance between NAS and connected computer was 271 km and this part of the world is generally known for having a good connection quality.

The installation process was relatively easy. However, we decided against trying to package the app properly for this initial test, as the purpose of the test was to see whether the hardware limits of the NAS would be an issue for speed or connectivity. Installing the application directly via the command line onto the Linux version already running on the device also proved challenging, as header files for libraries, etc. were missing. In the end we decided that the fastest way of arriving at our goal was to install a Debian *chroot* environment for which there was a community-built package available. The total package size of Fidus Writer was 235 MB at the time of the test, and even though the chroot environment needed to be installed as well, space was not an issue. The installation instructions provided with Fidus Writer[4] are written for Ubuntu 16.04, and these proved to be close to, but not the exact same as what was needed for Debian Jessie. Most notably did we need to install a newer version of Node.js than what the packaging system provided us with. We were then able to set the system up, connecting it even with a MariaDB database provided by another standard package on the NAS. The NAS was also able to reprogram the router to give us access to the port we decided to run our application on from the outside. The entire installation was done remotely without physical access to the NAS.

7 Test results

Our tests showed that at five clients could be connected to the NAS servers simultaneously and read/write the same document without any noticeable anomalies. We did not attempt to connect with more than five clients, as this number seemed more than sufficient for our purposes. While there were a few situations where the page would not load entirely the first time and it had to be reloaded, we attribute this

3 https://www.amazon.de/gp/offer-listing/B01BVDJGPE/ref=sr_1_1_olp?ie=UTF8&qid=1495437805&sr=8-1&condition=new

4 https://github.com/fiduswriter/fiduswriter/blob/3.1.0/README.md

to internet connection issues and not the NAS. The running of the combined Tornado/Django server for that makes out the backend of Fidus Writer and which is needed especially for the collaboration part, did not present a challenge to the NAS server.

Other parts of the editor – such as document export or import of citation sources in the BibTeX format – ran smoothly as well, but this was less of a surprise for us as we knew Fidus Writer had been programmed in client-heavy way, shifting most of the computing processes onto the client (browser) and only doing what is the minimum required on the server. Processes such as handling incoming document updates from clients and exporting/importing files are therefore implemented as much as possible in the browser and do not require server resources.

8 Conclusion

Academic document production continues to be challenging, especially when dealing with confidential material and when wanting to write in a way that preserves semantic information to avoid problems in the later stages of the publication process. Running open source semantic editing software is challenging because not everyone has access to their own server or can trust companies running such servers for them. An installation of a semantic editing software on a NAS server seems in many cases to be a viable alternative, as our tests running Fidus Writer on a Synology DS215J showed. Client-heavy applications such as Fidus Writer are well-suited for the job, as they will only require the minimal amount necessary from the NAS-servers and perform all other calculations in the browser. Packaging Fidus Writer as an app for the Synology system remains to be done before usability studies of the setup can commence.

This was our first investigation into the viability of a NAS-based solution. A survey on what kinds of editors would be interested in such a solution is among the items we are currently evaluating. The packaging of Fidus Writer as an app seems to be time-consuming and less of a technical challenge.

9 Acknowledgement

This work was funded by DFG, grant no. SU 647/19-1 and AU 340/9-1; the OSCOSS project at GESIS and University Bonn [3]. We would also like to thank other project participants and Fidus Writer participants who made the current version of Fidus Writer possible. Among these are Mana Azamat, Niloofar Azizi, Daniel Frebel, Babak Hashemi, Firas Kassawat, Takuto Kojima, Aleksandr Korovin, Fakhri

Momeni, and Anne Wittorf Kojima. Thanks also to Sarven Capadisli and Amy Guy for their thorough reviews[5] of an earlier version of this paper. Their comments have led to valuable additions in the current version.

10 References

1. Wilm, J., Frebel, D.: Real-world Challenges to Collaborative Text Creation. In: *2nd International Workshop on (Document) Changes: Modeling, Detection, Storage and Visualization*. p. 8:1--8:4. ACM (2015). http://dl.acm.org/citation.cfm?id=2723154&CFID=765669905&CFTOKEN=75305975 .
2. Afshin, S., Johannes, W., Philipp, M., Christoph, L.: Opening Scholarly Communication in Social Sciences by Connecting Collaborative Authoring to Peer Review. Information - Wissenschaft & Praxis. 68, 163 (2017). preprint: https://arxiv.org/abs/1703.04428 .
3. Mayr, P., Lange, C.: The Opening Scholarly Communication in Social Sciences project OSCOSS. Submitted. (2017). preprint: http://arxiv.org/abs/1611.04760 .

5 The open reviews of Sarven Capadisli and Amy Guy can be found at https://www.fiduswriter.org/2017/05/24/decentralized-creation-of-academic-documents-using-a-network-attached-storage-nas-server/#comments .