

Dealing with large data files using P2P collaboration software

Distributed Networks

Manuel Arjó

VicePresident Business Development
Sironta Software
Barcelona, Spain
e-mail: manuel.arjo@techideas.es

Jesús E. Gabaldón, PhD

Chief Scientific Officer
TechIDEAS
Barcelona, Spain
e-mail: jesus.gabaldon@techideas.es

Abstract—Using existing Delta encoding and Peer-to-Peer (P2P) technologies it is possible to edit and work with huge data files making an efficient use of resources. A new application tool, Sironta, which has been developed to improve the way to share and collaborate between partners by saving resources, thus not working on the entire files but on the modified parts of the documents and taking the maximum profit to the existing infrastructure.

This technological solution can be used to manage large files without overloading the whole infrastructure or collapsing it, because all the traffic is not centralized on a central server and we are not managing the entire documents. It also connects two or more users without making any network changes to share files of any type and any size from anywhere in the world without violating the terms of confidentiality or the NDA that such documents may have.

Sironta is a P2P collaboration tool with OpenOffice embedded to create, edit and exchange documents that require teamwork and to collaborate sharing these documents with other users.

I. INTRODUCTION

One of the main problems when dealing with large files is how to share this information. Nowadays, is not difficult to share lightweight voice, video, images or documents files. However, sharing large quantities of information is not easy when talking about documents. There's not even a common solution to work with people that is far away as if you where in the same room. The main problem is to share huge documents without spending all the bandwidth and resources on it.

Traditional web-based collaboration tools are usually moving the whole file all the time. When you need to change even one word in a large document of 30 Megabytes, you have to download it from the server, open it with the office suite, change the word, save the document and upload the 30 Megabytes again to the server. This is a waste of resources that most people and businesses can not afford. However, there are not many alternatives available.

II. OBJECTIVES

The principal objective of this research was to reach a solution for the large files management. It was not focused only on files management but also on the collaboration between partners working on the same documents or files from different locations.

The idea was to create a collaboration software that could manage large files without collapsing the network by cutting off the time and resources required for sharing a document.

III. METHODOLOGY

Traditional solutions are still mostly built upon strongly centralized hierarchies. Centralized systems may work well at small scale; as size increases, distributed systems can outperform centralized ones. Why?

Fundamentally, there is no direct exchange of information among peers in centralized networks. Therefore, all traffic passes through a server of highly centralized communications network. One of the main limitations of centralized networks is bandwidth. Since all data passes through central server, these are often under intense pressure. Consequently, providing sufficient bandwidth for several million users would demand burdensome investments in infrastructure and computer skills.

On the other hand, distributed and decentralized networks have neither servers or clients but nodes, and these can play both roles. The main goal of distributed networks is to use a large number of nodes with variable connectivity in unified form in order to minimize central organization. This is for three reasons:

- To avoid failure of a single point.
- To avoid resource limitations.
- To make more effectively use of distributed information and resources.
- To increase the performance and stability of the system.

Centralized computer networks were not intended to grow indefinitely and boundlessly. The growth rate of a computer network strictly follows the law of diminishing returns, which means that scalability costs are not linear but exponential in size. On the contrary, scalability costs remain

linear in distributed networks. In addition, the larger the number of nodes of the network, the greater its robustness and stability, as well as its total transport rate, service performance and efficiency. Consequently, there are at least a few powerful reasons for developing such networks.

Therefore, the best approach to create a collaboration network should be a distributed network. In this case, we chose a *Peer-to-Peer (P2P)* network. So, the idea that we had in mind was to write a *P2P* solution, as *Skype* or *Spotify* [1], but instead of sending sound we would send the differences between documents. The name of our solution is *Sironta*, a *P2P* collaborative tool used for exchanging, creating and editing documents that require teamwork.

This *P2P* approach allows every user having the information on his/her computer, so we can have a distributed repository and multiple backups not centralized on an external server. Even more, because of its *Peer-to-Peer* approach, we do not need to create a great infrastructure. We use users' network, systems and infrastructure, not a third-part infrastructure. It is not necessary to install additional servers for every new users, because the information is saved on their computers, not on a central server.

Thanks to *Peer-to-Peer* technology and *Delta encoding*, this solution has a perfect smooth management of large files. You can open the document inside the application and making the necessary changes in the document and synchronize. Only the changes are sent through the network, not the entire document. This resources saving implies that you can work with as many files as necessary, independently of their size.

Delta encoding (also known as *Delta compression*) is a way of transmitting data in the form of differences between sequential data rather than complete files. This technology is used by *Sironta* to send through the network only the changes made on a document (only the information required to obtain the new value from the older one is managed), saving a lot of network resources [2]. This is the reason why *Sironta* can manage large files without collapsing the net nor the system, because the modification of a large file does not mean great bandwidth usage.

Delta encoding works in this way: the differences between documents are recorded and consequently sent through the network in discrete files called *Deltas*. *Delta compression* greatly reduces data redundancy because changes are often small, and consequently the bandwidth used is significantly reduced if you compare it with sending the whole file. Non-encoded files are substantially less space-efficient than their groups of *Deltas* equivalents. The definition of *Delta* can be expressed as:

$$\Delta(v1, v2) = (v1 \setminus v2) \cup (v2 \setminus v1)$$

Where $v1$ is the first version of the document and $v2$ is the second version of the document, and they are successive.

The effectiveness of the compression algorithm is influenced by the nature of the data to be encoded. *Delta compression* performs best when data has constant or small variation; for an unsorted data set, there may be little compression possible.

Therefore, the larger the file is, the more useful is compared to web-based solutions.

To avoid transmitting errors to be considered as changes in the document, special error control code are used to detect them.

Delta encoding technology is also used to backup previous versions of all the documents changed inside *Sironta* [3]. This also reduces associated costs in the amount of data that has to be stored as differing versions. Thanks to this, *Sironta* has a *Document compare* function to see which changes has been made and who has made them.

Moreover, *Sironta* uses *GIT (Global Information Tracker)* to save the documents. *GIT* is a speed emphasis distributed revision control system [4]. *GIT* saves each new object created as a separate file. This system is completely inefficient because it takes a great deal of space. The solution used is to use packs that store, in a single file or in a network byte stream, a large number of objects using *Delta encoding* among themselves. To compress the packs the heuristic used is that files with the same name are, probably, similar, but do not depend on it for correctness. The newly added or created objects are still stored in a single way. To maintain the space-efficient storage, *GIT* does periodic repacking automatically.

Each revision of a file is stored as a unique blob object using *GIT*. By examining the tree and commit objects, the blobs relationships are found [5]. The new added objects are saved using *zlib* compression. But this is still a bad solution, because this kind of storage can quickly consume a large amount of space in the hard disk. For this reason, objects can be combined into packs, using *Delta encoding* to save space storing blobs as their changes relative to other blobs.

Sironta is open source and it is developed on *Eclipse*. It has a *GNU Affero General Public free software license (AGPL v3)*, based on the *GNU General Public License*, version 3 and published by the *Free Software Foundation* in November 2007 [6]). It uses *XMPP* protocol to manage the communications between peers.

Sironta is based on the room metaphor, first proposed by researchers at the *University of Calgary* in Canada on the article '*Heuristic evaluation of groupware based on the Mechanics of Collaboration*' in 2001 [7]. *Sironta* is divided into different rooms or workspaces in which every user can access a room and sharing the files he wants in that room, all the other users of the room can see this files and a user can access to different rooms or workspaces.

If you are not invited to a room, you can not enter in that room, so you can not see which documents are being created nor which files are being shared inside that room.

IV. RESULTS

The application has been tested by recreating the normal use. A large file of 100 Mb has been generated. Then it has been modified and transferred through the network. Time and resources measurements have been monitored to see the benefits of *Delta encoding* and *P2P* technology.

For the measurements, the port 5222 (*XMPP* client connection) of a wireless connection has been sniffed. In the first test, the whole 100 Mb file has been sent through the network using TCP, as a traditional web-based collaboration software would do. As a result, the transmission of the full document took 848,492 seconds.

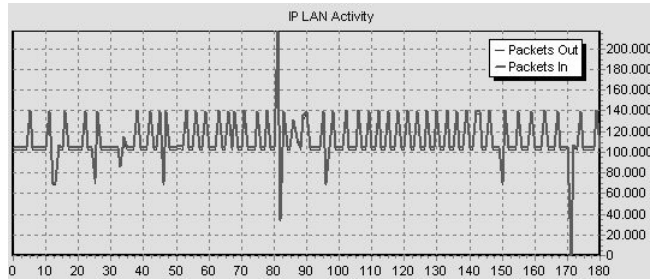


Figure 1. Bandwidth usage monitored in IP traffic bytes when transmitting a non-compressed 100 Mb file.

The second test realized has been to modify the same 100 Mb file and sent it again to the collaboration partners. Only 117 bytes has been modified in the file. In the traditional web-based collaboration software, the 100 Mb files would be sent again, so the results would be the same as before, 848,492 seconds. With the techniques described and *Delta encoding*, sending the changes to the other users took 34,923 seconds, including the whole room synchronization, with all the other documents in the room included.

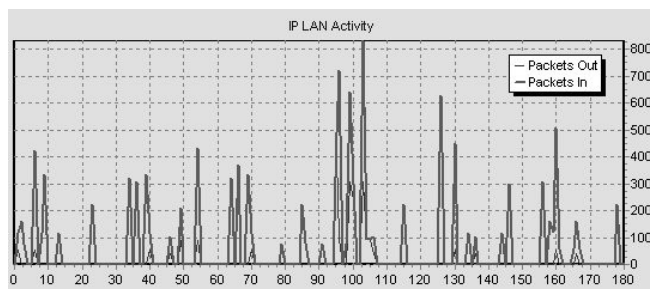


Figure 2. Bandwidth usage monitored in IP traffic bytes when transmitting a *Delta encoded* modification of a 100 Mb file.

As seen, the resources savings are near the 96 % using the *Delta encoding* in a 100 Mb file transmission. This is why users take the maximum profit of Sironta when working on large files.

V. FINAL STATEMENTS

The analysis has shown that this approach is an alternative to traditional collaboration software specially for large files. When working on a project, this technique is an optimal solution to save time to comply the terms of the draft.

Nevertheless, there are still some room for further improvements as a collaboration tool, adapting its functions to the results presented in this study.

Web-based solutions (not based on the *Peer-to-Peer* approach) may suit for students or single private works but they are not applicable to a company for security reasons.

Large-scale projects require a more elaborated infrastructure than centralized networks to maintain the privacy that these documents may have. Because of the *P2P* nature of the software, documents treated with this technique don't lose their confidentiality.

In the same way, large-scale projects often involve large-scale documents, and this is where *Delta compression* becomes indispensable to make collaboration possible.

Using *Peer-to-Peer* there's no need develop a new infrastructure for building the team network, because it uses the already existent infrastructure, and this is another kind of resource saving.

Taking advantage of the resource saving features, this collaboration tool should be an optimal solution for enterprises and companies working on huge projects with other partners.

ACKNOWLEDGMENTS

Sironta is encompassed within the *DEN4DEK* (*Digital Ecosystems Network of regions for DissEmination and Knowledge Deployment*) [8] and *OPAALS* (*Open Philosophies for Associative Autopoietic Digital Ecosystems*) [9] European projects.

REFERENCES

- [1] K. Molin, "Measurement and Analysis of the Direct Connect Peer-to-Peer File Sharing Network," Göteborg, Sweden: University of Gothenburg, June 2009.
- [2] T. Suel and N. Memon, "Algorithms for Delta Compression and Remote File Synchronization," Brooklyn, NY: Polytechnic University.
- [3] R. Burn and D. Long, "Efficient Distributed Backup with Delta Compression," Santa Cruz, CA: University of California.
- [4] S. Chacon and others, "GIT Community Book,".
- [5] C. García Campos, "GIT, a distributed Version Control System,".
- [6] "GPLv3 Third Discussion Draft Rationale," Boston, MA: Free Software Foundation, 2007.
- [7] K. Baker1, S. Greenberg, and C. Gutwin, "Heuristic Evaluation of Groupware Based on the Mechanics of Collaboration," Calgary, Canada: University of Calgary, 2001.
- [8] "The European Thematic Network on Digital Ecosystems," Regions for Digital Ecosystems Networks, June 2008.
- [9] "Open Philosophies for Associative Autopoietic Digital Ecosystems," Brussels: February 2009.