

Associative Way of Data Storage

Valery Kirkizh
Saint-Petersburg State University
of Aerospace Instrumentation
Saint-Petersburg, Russia
vkirkizh@vu.spb.ru

Vitaly Petrov
Tampere University
of Technology
Tampere, Finland
vitaly.petrov@tut.fi

Abstract

Ten years ago people have a problem how to save their digital information because data storage devices were expensive and had small capacities. Nowadays the cost of these devices has extremely decreased and they have very big spaces. So today users have got a new problem: how to find necessary file among a lot of different nested folders and files. Existed ways of file storages organizing have some insolvable issues, so the new way is proposed and described in this paper.

Index Terms: File system, Data storage, Tags.

I. INTRODUCTION

For the last ten years technological progress has led to an exponential growth of the information content created by people. About a quarter of a century ago computer users had an difficult problem: the amount of data exceeded existed capacities, and users weren't able to save all data which they wanted to save. That time data storage devices were very expensive, so only big companies, universities or research centers were able to get these devices. Also the capacities of that devices were very small and weren't able to save a lot of different important data, especially video. However, the development of information technologies made the cost of data storage very low (see Fig. 1) [1]. Now even an ordinary computer user can afford terabytes hard disks, blu-ray media discs [2], modern flash-drives with capacity of several tens and hundreds of gigabytes and cloud systems, which allow you to store a lot of information content. You can store millions and even billions of documents. But users have got the new problem. Among the huge number of different documents, files and media to find specific document (it can be a text file, image, video, letter, etc.) is very difficult task. Thus with amount of information increasing time which user usually spends for file searching increases too. Sooner or later this problem will be very important for each user in the world. The attempts to solve this problem have repeatedly been made by different researchers and engineers, but they weren't successful, because people tried to create existed storages structure more complicated.

II. MODERN FILE SYSTEMS STRUCTURE

Every digital storage device has its own file system. File system is the set of rules, which defines a way to organize, store and name of content [3]. Different platforms including personal computers and mobile devices use different file systems. The most popular among desktop file systems are NTFS [4], Ext3 [5] and HFS [6]. Also there are specialized file systems for certain devices, for example, well-known CDFS [7] for compact disks. But all of them are based on the most widespread data storage hierarchical method. This method, by-turn, is based on a hierarchical tree of directories

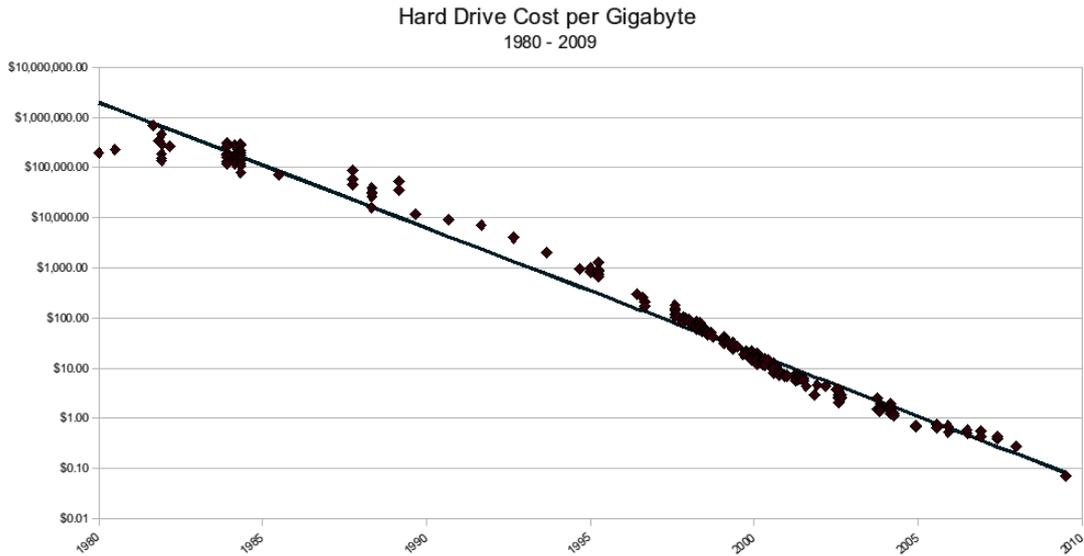


Fig. 1. Hard drive cost per gigabyte

constructing. Thereby each file has an unique path consisted of the directory name and the file name, which is an unique within the directory. This method has several disadvantages. The main of them is user freedom restriction. Other disadvantages are the redundancy of the information, the complexity of searching for the files, necessity to plan the directories structure in advance, etc.

Let consider mentioned disadvantages on the example. Some user has books and video materials on C++ and Java programming languages. If the user uses hierarchical way to store his files, he needs to create the following folders: «/Books/C++», «/Books/Java», «/Videos/C++» and «/Videos/Java». But also he may have other directories structure: «/C++/Books/», «/C++/Videos/», «/Java/Books» and «/Java/Videos». So, there is no convenient way to look at all videos or at all C++ materials and it is doesn't matter which structure user had chosen. The redundancy of the information is obviously, one way or another, some folders have duplicates with the same names. You can see an illustration for this example in Fig. 2.

Let consider another example. Some user has received a document (contract with company «Horns and Hoofs») via email and he needs to save it onto computer. There are some ways to choose the directory for this document:

- /Job/Contracts/Horns_and_Hoofs/contract1.pdf;
- /Job/Horns_and_Hoofs/Contracts/contract1.pdf;
- /Job/Horns_and_Hoofs/contract1.pdf;
- and many, many others.

It is necessary for user to choose directories structure in advance, when he doesn't have enough information about future files.

III. ASSOCIATIVE FILE SYSTEM

One of the solutions of these problems can be associative way of data storage. This way is based not on directory tree, but on the graphs of the tags. Each file has the set of tags instead of unique directory path. It allows to remove artificially created limitations of the hierarchical method [8].

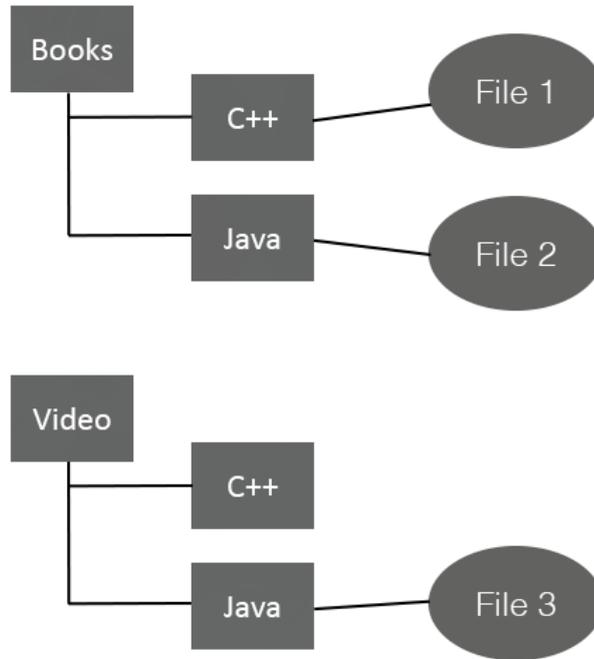


Fig. 2. One of hierarchical file structures for the first example

Consider the solution of the first example from previous section. You have books and video materials on C++ and Java programming languages and you have decided to use new method to store them. Now you have the only structure for your files. You give tags for your files. If you want to get all videos, all books, all C++ materials or all Java materials, you have to search them by one tag. If you want to get C++ books, you have to search them by two tags «C++» and «Books». Thereby habitual address bar becomes search bar. The order of file tags doesn't matter.

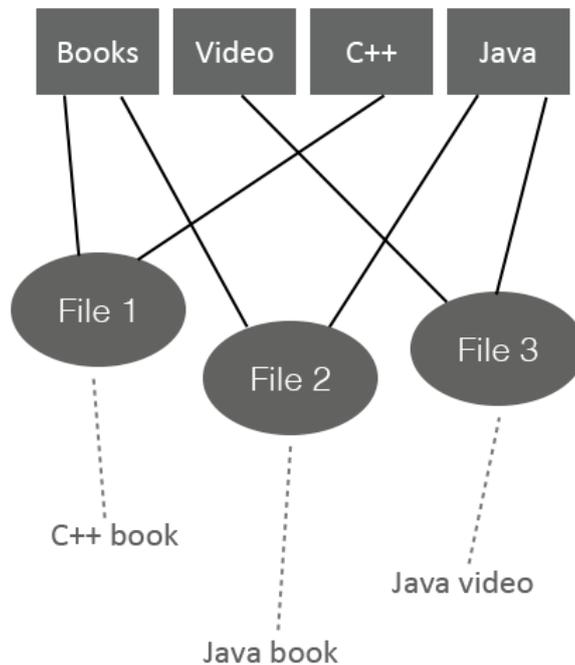


Fig. 3. Associative file structure for the first example

Consider solution of the second example. You have received a document (contract with company «Horns and Hoofs») via your email and you need to save it onto computer. And now you have the only way to do it: you have to give some tags to the document, for example, «Job», «Contracts» and «Horns and Hoofs». If you want to find all materials on «Horn and Hoofs» company, you have to search them by tag of the same name. If you want to get all contracts, you have to use tag «Contracts». If you want to look at all contracts with «Horn and Hoofs», you have to use both of these tags. You can see new file structure in Fig. 3. It is more convenient storage method than hierarchical one.

IV. EXISTED ASSOCIATIVE SOLUTIONS

Today there are several solutions for PC to organize the associative file system on your computer. Let consider the most popular of them. First solution is a file manager Elyse [9]. It is add-on for standard file systems for Windows and Mac OS X, it is an application for organizing your photos, videos and documents. You can see its screenshot in Fig. 4. It has a possibility to create nested tags. If you use Elyse, you cannot use tags in other programs. XYplorer [10] has the same lack and works only with Windows. There are a lot of such programs, but they are not interesting due to their using restrictions.

There are different web services, which allow you to save your documents, letters, notes, multimedia and other files on remote servers or cloud storages. For example, Gmail [11] is an email provider, which supports tags giving for your letters and further search by them. Gmail has a search-oriented interface and a conversation view similar to an internet forum. Letters can have file attachments, but you cannot give tags for them. Evernote [12] is a suite of software and services designed for note taking and archiving. A note can be a piece of formatted text, a full webpage or webpage excerpt, a photograph, a voice memo, or a handwritten ink note. Notes can have file attachments, but also without tags for files. Notes can be sorted into folders, then tagged, annotated, edited, given comments, searched and exported as part of a notebook. 4shared (former 4sync) [13] is a cloud storage for file synchronization between computers, mobile devices and web. It provides users with 15 GB for storage of pictures, music, video, documents and other types of files. 4shared provides hierarchical file structures with directories but allows to give tags for files and then operate with them.

Third class of the solutions are file systems, for example, XTagFS [14]. XTagFS is a FUSE [15] file system that organizes files and folders in Mac OS X using Spotlight Comment tags. Tags are represented as folders in XTagFS and tagged files are stored as links within them. The root directory of XTagFS shows all tags on your system as folders. Each tag folder contains the associated tagged files which are just symbolic links to the actual files on your Mac OS X file system. A tag folder also contains related tags as subfolders. For example, a file tagged with tags tag1 and tag2 can be accessed as both /tag1/tag2/filename or /tag2/tag1/filename. Another file system is py-tag-fs [16]. It is a developing user-oriented logical file system, which is above the current system-oriented file systems (like Ext3 and NTFS). Files will be organized with tags, instead of folders. Third considered file system is Tagged Virtual File System [17]. It is a high-level SOAP based file system for file sharing among Virtual Organization nodes in a grid. All these products are not full-fledged file systems, they are only add-ons for hierarchical file systems.

The comparison of mentioned solutions is given in the Table I.

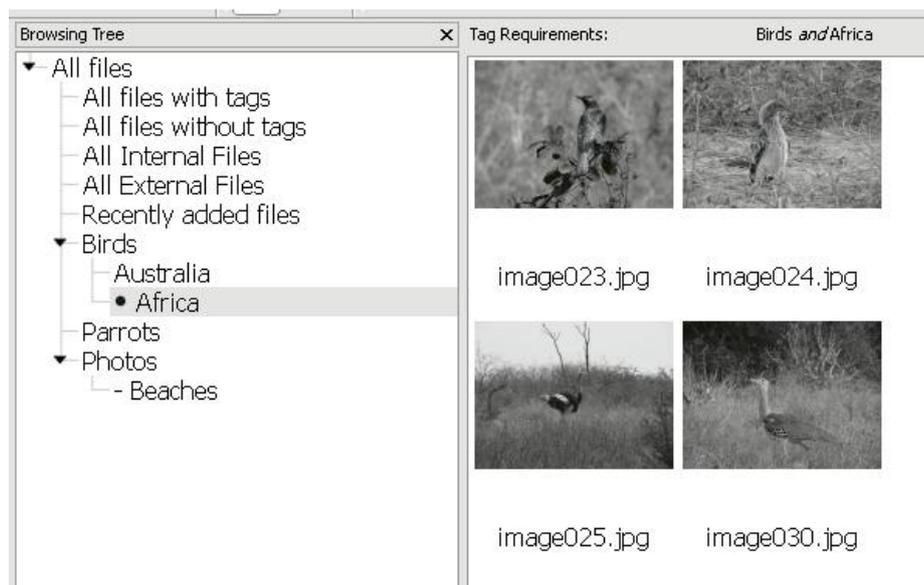


Fig. 4. Screenshot of Elyse file manager

TABLE I
EXISTED ASSOCIATIVE STORAGE SYSTEMS COMPARISON

	Elyse	XYplorer	GMail	Evernote	4shared	XTagFS	py-tag-fs	TVFS
Is an application	+	+	-	-	-	-	-	-
Is a web service	-	-	+	+	-	-	-	-
Is a file system	-	-	-	-	+	+	+	+
Nested tags	+	+	+	-	-	-	-	-
Multiple user access	-	-	-	+	-	-	-	+
Windows support	+	+	+	+	+	-	+	-
Mac OS X support	+	-	+	+	+	+	-	-
GNU/Linux support	-	-	+	+	+	-	+	+
Documents storing	+	+	+	+	+	+	+	+
Letters and notes storing	-	-	+	+	-	-	-	-
Multimedia files storing	+	+	-	-	+	+	+	+
All file types support	+	+	-	-	+	+	+	+
Using FUSE	-	-	-	-	-	+	+	-
Search bar	+	+	+	-	-	+	+	+
Query language	-	+	+	-	-	-	-	+
Pure associative storage	-	-	+	+	-	-	-	+

Unfortunately, these solutions have some problems connected with compatibility and productivity.

V. WEB-ORIENTED ASSOCIATIVE STORAGE

Associative way has not only a lot of advantages, but also it has some own issues. The main problem is a violation of the POSIX compatibility [18], which can lead to the necessity of reorganizing and rewriting the source code of a huge number of programs starting from application software and finishing of the low-level system services. Also associative storage method requires increased performance requirements.

Due to associative way issues the speed of its implementation on the PC is very slow, because you have to rewrite a lot of programs. Instead of this it is proposed to implement

this way in web-oriented environment. It will be cloud storage, which will allow to level difference between hierarchical and associative ways. But it will pure associative file storage. This service will provide remote file storage, files and tags management, API for third-party applications, web-interface, shared access to files and file groups with different access rights. One of the main research tasks in this project is multiple access providing. Developing service will be integrated with web-resources of the FRUCT Association, including the social network of the community. It will allow FRUCT members to keep their files and documents in cloud using tags.

VI. CONCLUSION

Traditional ways of data storage have a lot of problems, which become more and more noticeable every year. Quantity of using and storing information increases rapidly, so it is necessary to look for new data storage approaches. A new associative way for organizing data storage systems, which was described in this paper, is a good solution of many problems. There are some existing solutions based on this method, but all of them have different issues. Due to these facts web-oriented cloud associative file storage service implementation was proposed and started.

REFERENCES

- [1] Matt Komorowski, "A History of Storage Cost," <http://www.mkomo.com/cost-per-gigabyte>, 2009.
- [2] Blu-ray Disc Association, "The Blu-ray Disc Association Official Website," <http://www.blu-raydisc.com/>, 2004.
- [3] TechTerms.com, "File System Definition," <http://www.techterms.com/definition/filesystem>, 2007.
- [4] Microsoft Corporation, "New Technology File System," <http://technet.microsoft.com/en-US/library/cc737029.aspx>, 2005.
- [5] The Linux Information Project, "Third Extended Filesystem," <http://www.linfo.org/ext3fs.html>, 2003.
- [6] Apple Inc., "Hierarchical File System," <http://developer.apple.com/legacy/mac/library/documentation/mac/Files/Files-99.html>, 1996.
- [7] TechTerms.com, "Compact Disk File System," <http://www.techterms.com/definition/cdfs>, 2011.
- [8] Blog l'Ordikc, "Tag-based File System," <http://lordikc.free.fr/wordpress/?p=689>, 2010.
- [9] Silkwood Software, "Elyse File Manager," <http://silkwoodsoftware.com>, 2009.
- [10] Donald Lessau, "XYplorer," <http://www.xyplorer.com/tour/index.php?page=tags>, 2012.
- [11] Google Inc, "Google Mail," <https://mail.google.com/>, 2004.
- [12] Evernote Corporation, "Evernote Web Service," <http://evernote.com/>, 2009.
- [13] 4shared.com, "4shared Storage Sytem," <http://www.4shared.com/>, 2005.
- [14] Imran Patel, "XTagFS Project," <http://code.google.com/p/xtagfs/>, 2009.
- [15] Wikimedia Foundation Inc., "File System in Userspace," http://en.wikipedia.org/wiki/Filesystem_in_Userspace, 2011.
- [16] Zeemoo, "Tag-based File Sytem in Python," <http://sourceforge.net/projects/py-tag-fs/>, 2009.
- [17] Rodrigo Ruiz, "Tagged Virtual File System," <http://tvfs.sourceforge.net>, 2012.
- [18] IEEE Computer Society, "The POSIX Standart," <http://standards.ieee.org/develop/wg/POSIX.html>, 1988.