Interactive exploration of large filesystems

Joshua Foster, Kalpathi Subramanian, Gail-Joon Ahn

Joshua A. Foster, Kalpathi R. Subramanian, Gail-Joon Ahn, "Interactive exploration of large filesystems," Proc. SPIE 5669, Visualization and Data Analysis 2005, (11 March 2005); doi: 10.1117/12.589656

Event: Electronic Imaging 2005, 2005, San Jose, California, United States

Interactive Exploration of Large Filesystems

Joshua Foster[a] Kalpathi Subramanian[a] Gail-Joon Ahn[b]
[a]Computer Science [b]Software and Information Systems
The University of North Carolina at Charlotte, Charlotte, NC 28223, USA

ABSTRACT

Secure management of file systems of large organizations can present significant challenges to system administrators, in terms of the number of users, shared access to parts of the file system for supporting large software projects, and securing and monitoring critical parts of the file system from intruders. We present interactive visualization tools for monitoring and viewing the complex access control relationships within large file systems. This tool is targeted as an aid to system administrators to manage users, software applications and shared access. We tested our tool on UNC Charlotte's Andrew File System (AFS), which contains 7043 users, 560 user groups, and about 2.1 million directories. Our system displays summary information about the file system, and two types of visualizations to explore access control relationships among classes of users. In addition, drill-down features are provided to explore the user file system structure and manage access control information of any directory within the system. All of the views are linked to permit easy navigation and features are provided that make the system scalable to larger filesystems.

Keywords: AFS, filesystem, monitoring, drill-down, visualization

1. INTRODUCTION

Current trends in electronic information storage, communication and exchange of business documents, facilitated in large part by the Internet, inexpensive disk storage, etc., has resulted in massive file systems in large academic and industrial organizations. Administration and secure management of such file systems is a challenging task. System administrators have to typically manage large numbers of users and software applications, while at the same time ensuring the security of critical parts of the file system, privacy of sensitive information, and ensure appropriate access to users. While some of these issues are also the domain of file systems, the sheer size and the increasing risk of intruders demands more robust and scalable solutions. Information visualization techniques can be employed to address many of these challenges to assist the system administrator.

The majority of the work on file system visualization has focused on utilizing their size attribute. Treemaps[1] portray a file system using a compact space-filling(pixelized) representation, by recursively subdividing a 2D rectangular region into partitions as a function of the subtree file/directory size. To overcome some of the difficulties (such as poor aspect ratio) of this representation, several refinements have been proposed. Ordered treemap layouts[2] provide more stability for dynamically changing data as well as better aspect ratios, cushion treemaps[3] use shading to bring out the hierarchical structure, and beam trees[4] use nested cylindrical beams to display hierarchical information.

Work on utilizing visualization to monitor network intrusion detection[5] has also been explored by researchers. Erbacher[6, 7] analyzed log files of campus network activity of their university's principal server and a dozen other workstations. The visualization consists of circular glyphs, with the main server at the center and nodes connected to the server placed radially depending on their IP addresses, as well as whether they are a local or a remote client. Links between clients and servers may be colored, directed, or dashed, representing various attributes of the connection. Similarly, Teoh et al.[8] built visualization tools to examine Internet routing data, which can also become very extensive. The goal here was to detect and comprehend anomalous events. They used a pixelized visualization based on a quadtree subdivision of IP prefixes. Links representing routing paths of interest were visualized over time, in order to determine their stability.

Correspondence: Kalpathi Subramanian, krs@uncc.edu
Author contact information: Email: {krs,jafoster,gahn}@uncc.edu

Visualization and Data Analysis 2005, edited by Robert F. Erbacher,
Jonathan C. Roberts, Matti T. Gröhn, Katy Börner, Proc. of SPIE-IS&T
Electronic Imaging, SPIE Vol. 5669 © 2005 SPIE and IS&T · 0277-786X/05/$15

While the work presented here is not directly related to network security, a long term goal of this research is to monitor and detect in real-time file system intrusions; in particular, the existence of trustworthy log files is of importance in computer forensic analysis.[9] Log files contain valuable information that may be destroyed by sophisticated attackers to remove any record of their activities after an intrusion. Such critical parts of large file systems may be specially flagged for proactive monitoring and detection.

In this article, we present visualization tools to explore user and access control relationships within large filesystems. The current system uses well understood visualization techniques to display relationships: hierarchical displays, pixelized visualizations, zooming and linked views. It displays relationships across different classes of users, displays file/user access vulnerabilities from a security standpoint, displays a selected user's file structure, and access privileges of any particular directory within. Access privileges of a chosen directory may be modified, either interactively or via file system specific commands. A graphical interface provides the basis for easy navigation, with drill down features to points of interest. The different views in our system are *linked*, so as to maintain context. We demonstrate our visualization system using an Andrew File System (AFS)[10] dataset that was acquired from our campus network. A number of metrics have been designed to define "interesting" users and incorporated within the visualizations. These metrics support exploration of the file system and facilitate specific visual queries, and is expected to be of assistance to system administrators.

2. METHODS

2.1. Data Acquisition

We demonstrate our file system visualization system using an Andrew File System (AFS)[10, 11] dataset. AFS is a highly scalable filesystem that has found wide acceptance in large academic and industrial organizations. AFS imposes access control mechanisms (which is of interest here) in all of its directories, in a manner that promotes flexibility. Similar to Unix, it supports user groups, however, users themselves may create groups (in contrast to just system administrators) and assign privileges. Each AFS directory contains seven access control privileges: read, lookup, insert, delete, write, lock and administer. Our initial goal was to extract all these relationships from a large filesystem and support interactive visual queries.

The file system structure was extracted (using scripts) from our campus network and saved to disk, after masking user names and other personal information. Filesystem data for the system came from two sources: a listing of all groups and their members, and a recursive listing of the access controls for every directory in the system. This data can be retrieved using simple AFS commands. The data totals about 450 MB in text form, representing, 7043 users, 560 user groups and about 2.1 million directory names and their access privileges. We used a converter to read in the files, build in memory the data structures needed by our system, and save these structures in binary form. The resulting binary data file was about 80MB.

Users were classified into faculty, students, administrators and dormant (inactive users). In addition, a set of critical directories were identified for specialized processing. These directories typically contain software applications that are of general use by larger numbers of users. In our system, we group these together as a distinct class of *critical* users.

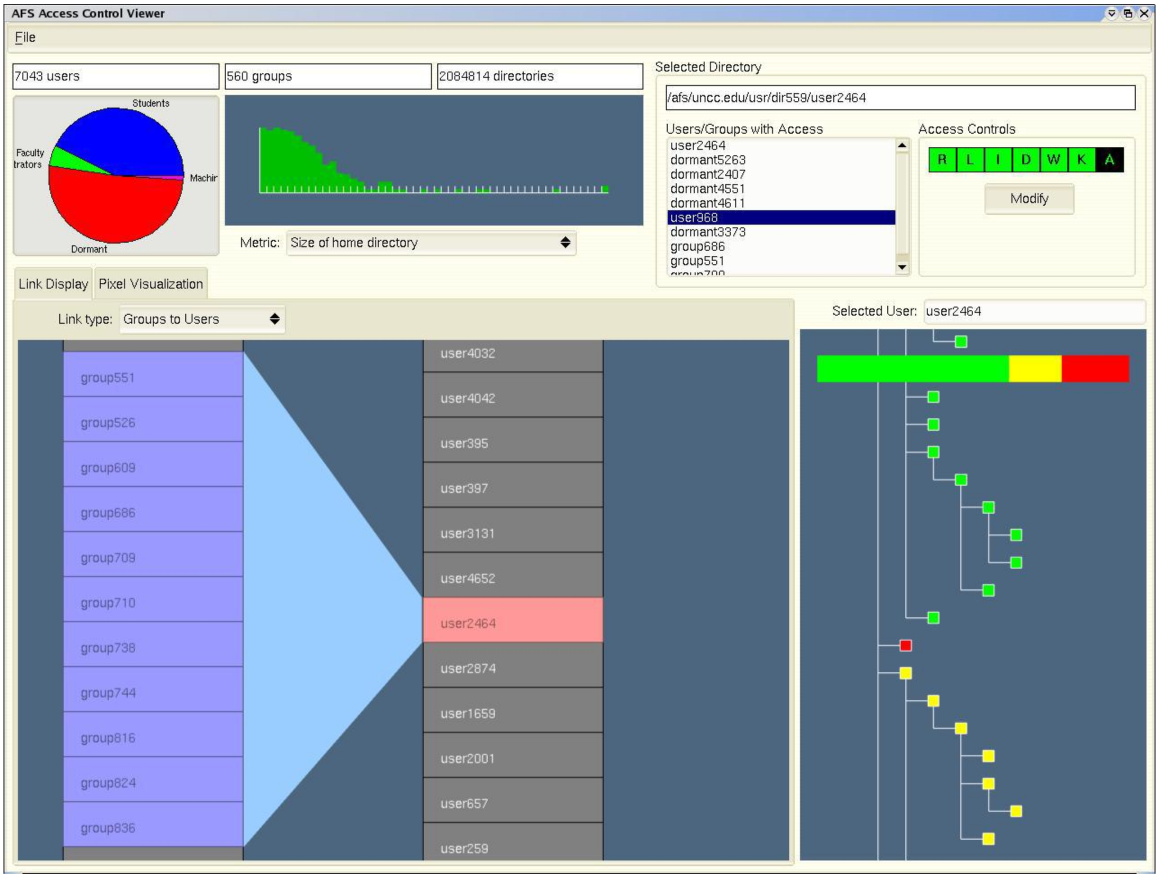
2.2. Data Structures

There are three types of entities in the visualization system; therefore three main data structures: *User, Group, and Directory*. A *Group* simply contains a list of references to the *Users* who belong to it. *Users* consist of a reference to their home directory, and a *Statistics* structure which is a catch-all for various statistics computed about the user and his/her directories. Each directory is a node in a tree structure, containing a list of references to its children. Access data is also carried at the directory level. Each directory contains two lists of *AccessRights* structures: one for user access rights and one for group access rights. The *AccessRights* structure simply contains a reference to the corresponding user or group and an access byte, each bit of which represents one type of access: read, look, insert, delete, write, lock, and administer. At the top level, there is a master list of groups and a master list of users, with each user having a reference to the top node of a tree of *Directories* representing the path to the home directory. The organization of the filesystem data in memory parallels the drill-down method of data exploration in the visualizations; that is, selection of a user or group first, then selection of one of the user's directories, and finally selection of an individual group or user with access to that directory.

2.3. Visualization Design

The file visualization system was designed with the following considerations:

- Support both high-level overviews as well as detailed structure of the filesystem.
- Smooth interactive navigation across different levels of detail, providing the means to drill-down into features of interest.
- Highly scalable, in terms of the number of users, and files/directories.
- Support metrics for constructing queries to identify features of interest, as well as encourage interactive exploration of the dataset.
- Robust and secure management of users and file system access.


AFS Access Control Viewer

7043 users

560 groups

2084814 directories

Selected Directory: /afs/uncc.edu/usr/dir559/user2464

Users/Groups with Access:
user2464
dormant5263
dormant2407
dormant4551
dormant4611
user968
dormant3373
group686
group551
group709

Metric: Size of home directory

Link type: Groups to Users

Selected User: user2464

Figure 1. Relationships between classes of users

Fig. 1 shows our overall design. The upper left panel shows aggregate information of the filesystem, the lower left panel displays user-centric relationships (two different visualization types are currently supported), the lower right panel illustrates user file system structure, and the upper right panel shows access control privileges and associated information for a particular user or directory. We describe each of these views next.

Aggregate View

The high-level visualizations show a summarized view of the file system. There are four classes of users: students, faculty members, administrators, and dormant accounts. A pie chart illustrates the distribution. The total number of users, groups and directories are also displayed. The histogram view reflects the distribution of the users, in terms of three important metrics: size of the user's home directory (defined as the total number of directories under the user's home directory), number of groups the user belongs to, and percentage of directories the user is *sharing* (we define a *shared directory* as one for which the owner has given access privileges to another user or group). For instance, it can be seen in Fig. 1 that a majority of users own small sized home directories (which is generally to be expected in very large academic filesystems). This could be useful in thresholding such users, in order to identify "interesting" users; for instance, by thresholding users owning a very small number of directories.

User Relationships View

This view permits exploring relationships between classes of users as well as displaying user's relationships to underlying attributes of their file system. Two visualization types are currently supported, (1) scrollable lists of users with explicit links to display relationships, and (2) a pixel style visualization that can compactly represent all users, suitably colored to reflect a linear combination of various attributes of interest.

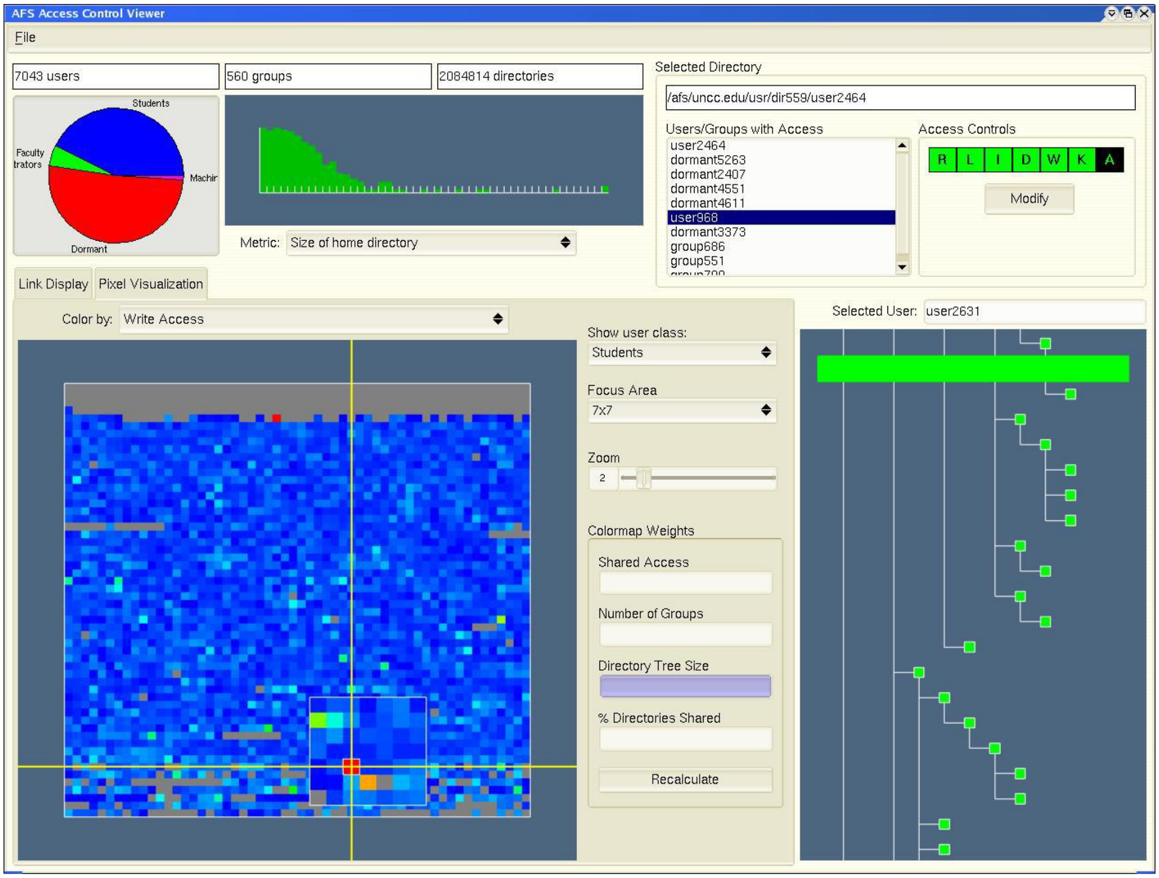
The link display (lower left panel, Fig. 1 reveals the relationships between students, faculty, and groups. This visualization consists of two vertical scrollable lists, either of groups and students or faculty and groups. When an item in the scrollable list is selected, its associated entities in the second list are linked to and highlighted. For example, one can switch to the groups/students display and select a user, and all groups that the user belongs to will be highlighted. One of these groups may then be selected and the visualization switched to faculty/groups mode to see all the faculty members that belong to that group. Linked items are highlighted, clustered together, and automatically scrolled to the center of the window, and a linking polygon is drawn to connect them. The lists may be manually scrolled using the mouse wheel, Page Up/Down keys, or by moving the mouse towards the top or bottom of the list. In Fig. 1, user2464 is seen to belong to 11 different groups. The panel on the right automatically also displays his home directory structure (as described later).

A pixel-style visualization is also provided to display additional relationships relating to the underlying user file system. Users are represented within a square grid of cells; currently the users are ordered by their id, which is determined by the input dataset. Any of the four user classes (student, faculty, dormant, admin) may be shown individually. The cell for each user is colored according to a weighted combination of four parameters,

- Size of the users' home directory,
- Number of groups the user belongs to,
- Percentage of directories the user shares with other users/groups.
- Type of access in shared directories.

The first 3 parameters have been mentioned earlier; the fourth parameter distinguishes users based on the access privileges provided by a user to other users or groups. For example it may be important to classify the users on the basis of how much read/look, write, or administrative access they have given other users or groups to their directories. Each of these four parameters has an importance, or a weight, which may be adjusted with sliders. The final color for each user is a normalized sum of the values for each of these parameters multiplied by the parameter's weight. Colors are mapped from blue to red, and cells with a final value of zero (no importance with the selected parameters and weights) are colored grey.

Fig. 2 illustrates pixel visualization of the students class. In this example, the visualization displays the size of the student directories. The cross-hairs show the current selection (the red square), which is user2631 with 4476 directories. While this case could have been better illustrated with a treemap visualization, other attributes can be mapped (as we will see later) to the user glyphs.



AFS Access Control Viewer

7043 users

560 groups

2084814 directories

Selected Directory: /afs/uncc.edu/usr/dir559/user2464

Users/Groups with Access:
user2464
dormant5263
dormant2407
dormant4551
dormant4611
user968
dormant3373
group686
group551

Metric: Size of home directory

Color by: Write Access

Show user class: Students

Focus Area: 7x7

Zoom: 2

Selected User: user2631

Figure 2. File system relationships using pixel style visualization

While all of the users (in this example dataset) are easily accommodated in the visualization, larger datasets can cause difficulties in interactive user selection. Thus, we provide a *magnifying glass* or local zoom capability as part of the interface. The local zoom is centered around the cursor position and its level and size of the magnified area are both adjustable. As can be seen in Fig. 2, all the users within the white box surrounding the selected(red) user are magnified for ease of picking. Currently, areas outside of the zoom are of the original size; in the future, we could consider adjusting their size based on their distance from the focus area, similar to normal focus+context methods.[12]

User File Structure View

When a user is selected in either the link display or the pixel visualization, the user's home directory tree is shown in the directory display (lower right, Figs. 1, 2). The directory structure is shown with a Windows Explorer style vertical layout. Each directory is represented with a colored square, either red, green, or yellow. Non-shared directories are green, and shared directories are yellow or red depending on the type of sharing. A yellow directory represents one in which the owner has given access to either another user or another group, but not both. Directories in which the owner has given access to other users and other groups are red. A horizontal bar at the top of this panel displays the distribution of the shared directories for a quick summary. This visualization supports trackball-style panning and zooming and is scrollable with the mouse wheel. One difficulty with this node-link layout is its inefficient use of screen space and thus even moderately sized user directories will easily exceed the boundaries of the view. An alternate is to use a treemap style display, for instance, by using a count of subdirectories as a metric for the partitioning (if directory/file sizes are not available).

Access Control View

At the finest level of detail, information about a selected directory may be shown. Selecting any directory within a user's file structure results in the the directory's access privileges being displayed (upper right panel, Figs. 1, 2). The full pathname to the directory is displayed at the top, along with a list of users and groups with access to the directory. Any of these users and groups may be selected to reveal the individual access privileges given to that user. In Fig. 2, user2464 is sharing the displayed directory with a number of users and groups; user968's access privileges are displayed. The individual access privileges of a directory may be modified at this step, as this is a tool that will be geared toward filesystem administrators in the future.

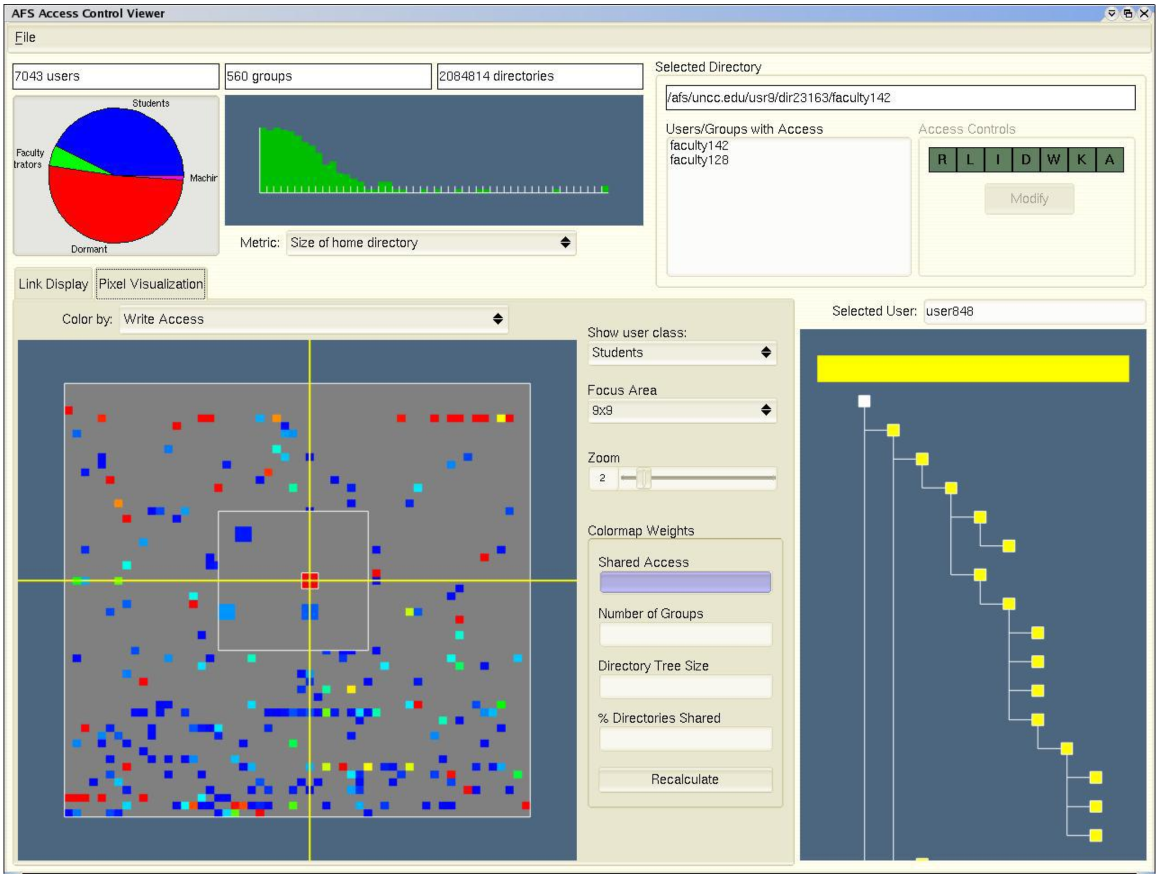
3. RESULTS

3.1. Implementation

We are developing this system in C++ under Linux*. All of the drawing is done in OpenGL, using the FLTK toolkit[13] for the user interface.

3.2. Examples

We describe three examples to illustrate the features of this tool as well as its possible application for filesystem monitoring. As this system is currently a prototype, no extensive testing or user study has been performed yet.


AFS Access Control Viewer

7043 users

560 groups

2084814 directories

Metric: Size of home directory

Selected Directory: /afs/uncc.edu/usr9/dir23163/faculty142

Users/Groups with Access: faculty142, faculty128

Color by: Write Access

Selected User: user848

Show user class: Students

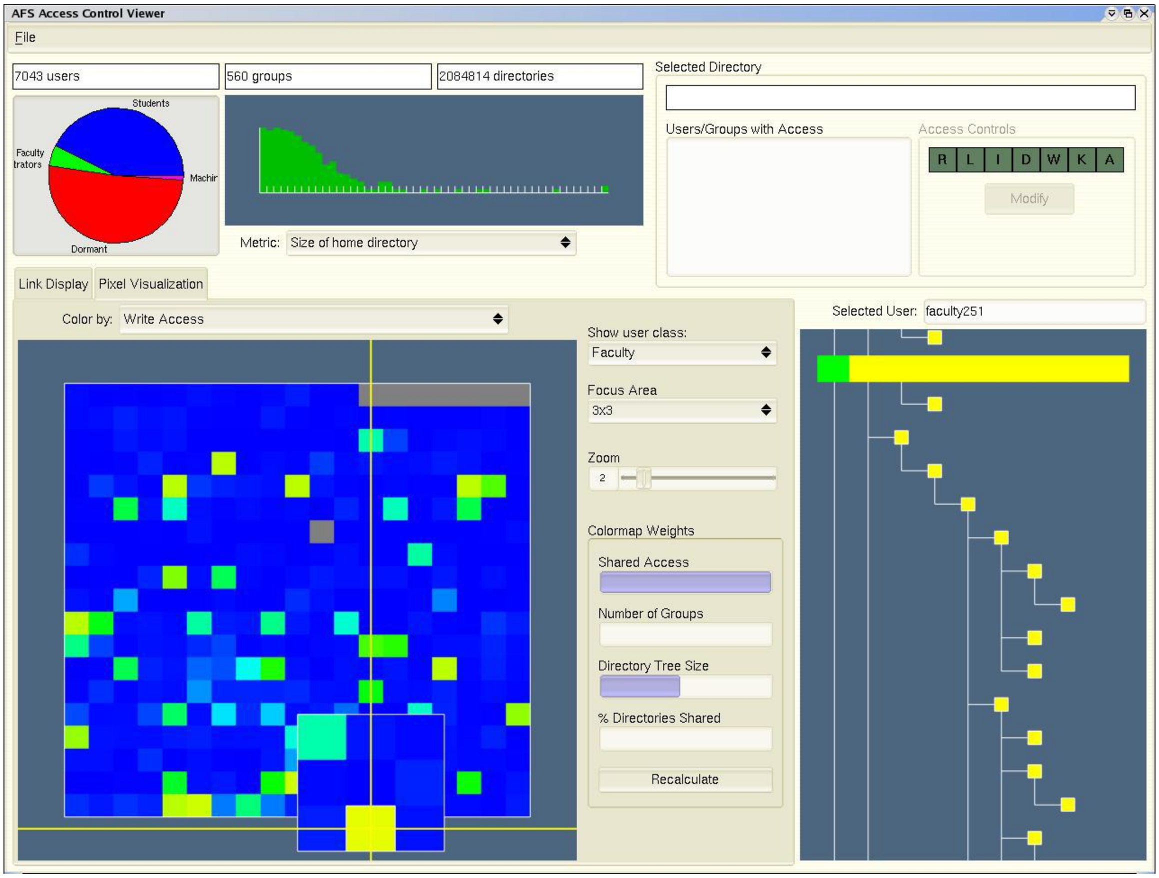
Focus Area: 9x9

Zoom: 2

Figure 3. Example illustrating student users sharing write privileges

*As we use public domain tools, this application can be easily ported to Windows based PCs or other flavors of Unix.

Fig. 3 illustrates student users in the pixel visualization. Here the color mapping illustrates the Shared Access attribute specific to Write Access; in other words, we are interested in looking at users who have given write access to other groups or users. As all other attributes have been turned off, the colormap is determined by the percentage of user directories that are writable by other users. The selected user in Fig. 3, user848 has 193 directories under his home directory. The yellow horizontal bar in the lower right panel clearly indicates the 100% sharing for write privileges. It is possible that a user might have mistakenly provided such privileges; this might alert an administrator (with a tool such as this) to warn the user. More important, this visualization indicates that there are a number of users (red squares indicate almost the entire directory tree is writable by at least one other user or group) who have provided write access privileges to other users/groups.


AFS Access Control Viewer

7043 users

560 groups

2084814 directories

Metric: Size of home directory

Color by: Write Access

Show user class: Faculty

Focus Area: 3x3

Zoom: 2

Selected User: faculty251

Figure 4. Example illustrating faculty users sharing write privileges in combination with home directory size

Fig. 4 indicates the pixel visualization of the faculty users. Here again, we explore the Shared Access attribute with Write access; however, we now make this sensitive to the size of the directory. Thus the directory size attribute (slider) is turned on about halfway. To reach a high value (red square), a user must have a large directory, as well as provide Write access to a significant part of his/her home directory. The picked user (yellow square in Fig. 4), faculty251 has 5400 directories and shares 89% of the filesystem with other users.

Fig. 5 illustrates an example of modifying user access privileges. Here faculty245's home directory has full administrative access to user657 except for directories. The top panels show the selection of faculty245 (orange square in the top left panel), and the largely yellow bar and yellow squares in the directory view(top right panel).

We have implemented the means to apply file system commands to our data in memory (at this time); the bottom panels indicates the result after user657's access privileges have been removed. Now faculty245 is the blue square, indicating that this user's directory is largely unshared. This is confirmed by the mostly green

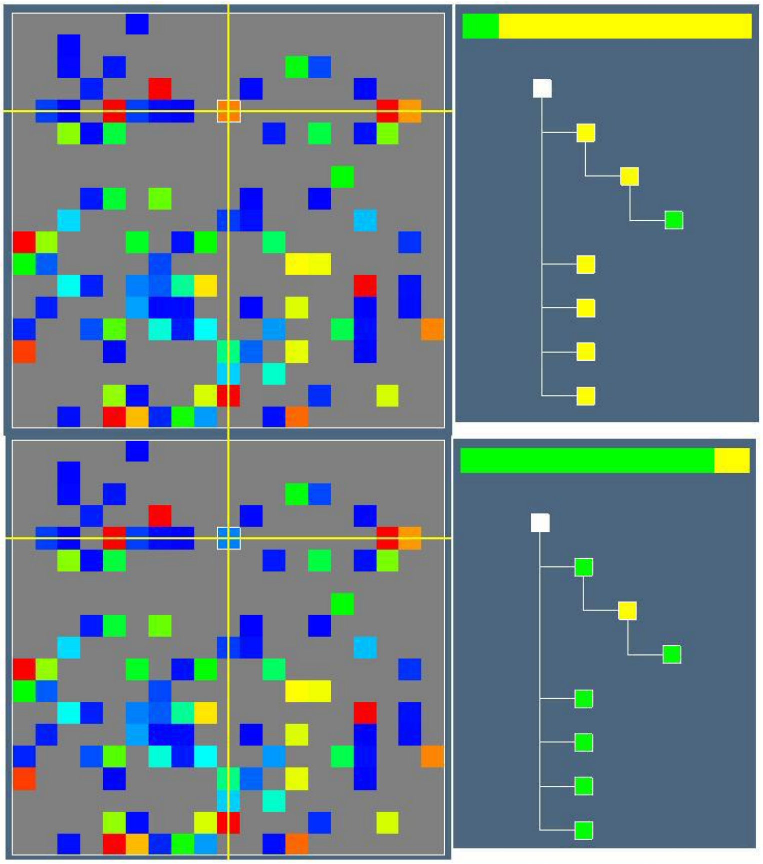


Figure 5. Example illustrating modification of access control privileges, before(top panels) and after (bottom panels).

horizontal bar and the green squares in the directory view. This simple example illustrates a more robust and less error-prone means to managing access privileges of users who might have very large home directories.

4. CONCLUSIONS

As filesystems continue to increase in complexity and size in the information age, secure management and monitoring of such systems will be an important issue in large-scale networks. In this article, we have presented scalable information visualization tools that can provide a front-end to such large systems. By looking at different attributes of user directories, we have begun to define useful metrics that can capture the underlying relationships. Here we have demonstrated our system with the AFS filesystem.

There are a number of issues that need to be dealt with before this system can be utilized by system administrators on a routine basis, as follows:

- Currently, our system works with a *snapshot* of the file system. Changes to the file system and access privileges do not update the visualization system. Thus, we will need the means to record such changes to both file system structure (creation and deletion of users, directory changes, etc.), and access privileges need to be recorded in log files that can be monitored by the visualization system, so as to be uptodate. Given that such changes come from a large body of users and groups of system administrators, this brings up issues of synchronization, that must be addressed.
- Changes to the access privileges (Fig. 3) are only made to the data structures in memory, to illustrate the capabilities of the system. These changes should then be propagated to the underlying file system. The

ability to change access privileges by the administrator via the visualization interface is a more attractive means to deal with large filesystems.

- The current system is somewhat user-centric, meaning only user directories are stored. However, there are some non-user directories that must still be monitored, such as applications and data directories. In the example dataset, there are twelve such *critical* directory paths that can be monitored. In the future, we would need to further classify these into individual application directories. For instance, during upgrades, these directories may undergo significant changes in structure and user access and the visualization system would be useful to ensure that the access privileges are appropriate prior to public release.

Our long-term goal is to use information visualization as the means to view and comprehend complex security policies that are currently in use, as well as understand their vulnerabilities and shortcomings. This can then lead to the development of new infrastructure and strategies that call for shared access to resources in networked environments. Mechanisms must be provided to protect sensitive and confidential information from adversaries. We believe our tool can address the issue of how to advocate selective information sharing while minimizing the risks of unauthorized access through the effective visual analysis of a) unauthorized sharing and b) violation of access control policy.

5. ACKNOWLEDGEMENTS

Our thanks to Dr. Charles Price who generated several versions of the AFS filesystem dataset. The work of Gail-Joon Ahn was partially supported at the Laboratory of Information Integration, Security and Privacy at the University of North Carolina at Charlotte by a grant from National Science Foundation (NSF-IIS-0242393) .

REFERENCES

1. B. Shneiderman, "Tree visualization with tree-maps:2-d space filling approach," *ACM Transactions on Graphics* **11**(1), pp. 92–99, 1992.
2. B. Shneiderman, M. Wattenberg, and D. Jones, "Ordered treemap layouts," in *Proceedings of IEEE Information Visualization 2001, Oct. 22-23, San Diego, CA.*, pp. 73–78, IEEE Computer Society, 2001.
3. J. van Wijk and H. van de Wetering, "Cushion treemaps: Visualization of hierarchical information," in *Proceedings of IEEE Information Visualization 99, Oct. 24-29, San Francisco, CA.*, IEEE Computer Society, 1999.
4. F. van Ham and J. van Wijk, "Beam trees: Compact visualization of large hierarchies," in *Proceedings of IEEE Information Visualization 2002, Oct. 19-24, Boston, MA.*, pp. 93–100, IEEE Computer Society, 2003.
5. P. Proctor, *Practical Intrusion Detection Handbook*, Prentice Hall Inc., 2000.
6. R. Erbacher, K. Walker, and D. Frinckle, "Intrusion and misuse detection in large-scale systems," *IEEE Computer Graphics and Applications* **22**, Jan/Feb 2002.
7. R. Erbacher, "Visual traffic monitoring and evaluation," in *Proceedings of the Conference on Internet Performance and Control of Network Systems II*, pp. 153–160, Denver, Co., August 2001.
8. S. Teoh, K. Ma, and X. Zhao, "Case study: Interactive visualization for internet security," in *Proceedings of the IEEE Visualization 2002*, pp. 505–508, nov 2002. 0ct. 27-Nov. 1, Boston, MA.
9. Y. Wang and Y. Zheng, "Fast and secure worm storage systems," in *Proceedings of the IEEE Security in Storage Workshop (SISW)*, pp. 11–19, Oct. 31, Washington DC 2003.
10. R. Campbell, *Managing AFS, The Andrew File System*, Prentice Hall Inc., 1998.
11. "Open afs." http://www.openafs.org.
12. Y. Leung and M. Apperley, "A review and taxonomy of distortion-oriented presentation techniques," *ACM Transactions on Computer Human Interaction* **1**(2), 1994.
13. B. Spitzak, "The fast light toolkit." http://www.fltk.org.