Interactive Exploration of the AFS File System

Joshua Foster*
Computer Science

Kalpathi Subramanian†
Computer Science

Robert Herring‡
Computer Science

Gail Ahn §
Software and Information Systems

The University of North Carolina at Charlotte, Charlotte, NC 28223

ABSTRACT

Managing file systems of large organizations can present significant challenges in terms of the number of users, shared access to parts of the file system, and securing and monitoring critical parts of the file system. We present an interactive exploratory tool for monitoring and viewing the complex relationships within the Andrews File System (AFS). This tool is targeted as an aid to system administrators to manage users, applications and shared access. We tested our tool on UNC Charlotte's Andrews File System (AFS) file system, which contains 4554 users, 556 user groups, and 2.2 million directories. Two types of visualizations are supported to explore file system relationships. In addition, drill-down features are provided to access the user file system and access control information of any directory within the system. All of the views are linked to facilitate easy navigation.

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

1 INTRODUCTION

Administering and managing large file systems is becoming an important and challenging task, and has been partly driven by the explosion of electronic information, inexpensive disk space, etc. System administrators have to typically manage such large systems, across 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. 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[3] portray the file system as a compact 2D 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 of this representation, several refinements have been proposed. Ordered treemap layouts[4] provide more stability for dynamically changing data as well as better aspect ratios, cushion treemaps[6] use shading to bring out the hierarchical structure, and beam trees[2] use nested cylindrical beams to display hierarchical information.

In this work, we present an interactive visualization tool to extract and visualize important file system relationships across different classes of users, look for possible vulnerabilities from a security standpoint, and provide an intuitive means to navigate and drill down to points of interest. The immediate goal is aimed at a system that can be indispensable to the system administrator, while in the

---

*e-mail:jafoster@uncc.edu
†e-mail:krs@uncc.edu
‡e-mail:rherring@uncc.edu
§e-mail:gahn@uncc.edu

long term this can be deployed across very large file systems over enterprise or corporate networks.

2 METHODS

2.1 Data Acquisition

We demonstrate our interactive visualization tool using an Andrews File System (AFS)[1] example. The file system structure was extracted from our campus network and saved to disk, after masking user names and other personal information. The extracted file system (about 450MBytes)contained 4554 users, 556 user groups and about 2.2 million directories and their access privileges. Users were further classified into faculty, students (named 'users' in the data file) and dormant (inactive users). Each AFS file/directory contains seven access control privileges: lookup, read, insert, delete, write, lock and administer.

2.2 Visualization Tools

The AFS Access Control Viewer[1] loads binary files containing a listing of all directories, groups, users, and access control lists for the filesystem. As shown in Fig. 1, the viewer has 4 panels:

- *Aggregate View (upper left):* Using a pie chart, shows the distribution of students, faculty, administrators and dormant users.

- *User Relationships (lower left):* This panel permits relationships between different classes of users to be displayed explicitly. Currently we support two types of views, (1) scrollable lists of users, (2) pixel type visualization.

- *User Directory (lower right):* This view shows a selected user directory structure, using an Internet Explorer style view.

- *Access Control Privileges(upper right):* These are explicitly shown for the selected user (multiple users/groups may have access to a directory) and may be modified.

Fig. 1 shows an example of viewing relationships using a pair of linked scrolling lists. These lists can be configured in different ways, for example relating groups to users or relating faculty to groups. When any item in a list is clicked, the corresponding items in the other list is highlighted. For example, clicking on a group highlights all the users in that group; and conversely, clicking on a user will highlight all the groups the user belongs to. The lists can become quite large, so scrolling is supported in three ways: using the mouse-wheel, PageUp/PageDown keys, or by moving the mouse towards the top or bottom of the list. A linking band connects the selected item in one list to the corresponding items in another list. In Fig. 1, we have selected "user7", who is part of 7 other groups. The user's home directory structure is shown to the right (zoomed in). The white icon in the user's file system has been selected, updating the list of users having access to this directory.

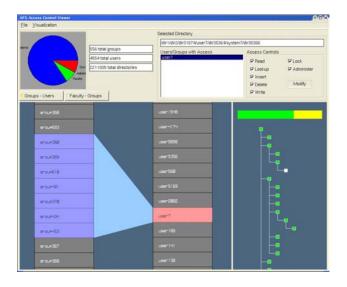[1]A live demo is planned for the poster at the symposium.

Figure 1:

In this case, the user7 has sole ownership and selecting this button displays the access control privileges on the upper right view.

Alternately, a pixel visualization can also be selected. This has the advantage of accommodating all users within the view and is thus very compact and space-filling, in a manner similar to treemaps. We represent each user as a colored block in a grid. In this view, the color of a block represents a 'warning level' for the corresponding user, computed by one of several selectable functions. As an example, one function calculates the warning level of a user as the number of directories owned by that user while at the same time they are accessible to additional users or groups. Fig. 2 shows an example. Here black regions indicate users who have sole ownership (no directories are shared), shades of green to red indicate increasing levels of ownership by other users/groups. We have selected the user corresponding to the red cube (lower right, Fig. 2), who shares 2855 directories with other users/groups. The user's file system and access privileges of a shared user is also displayed, as before.
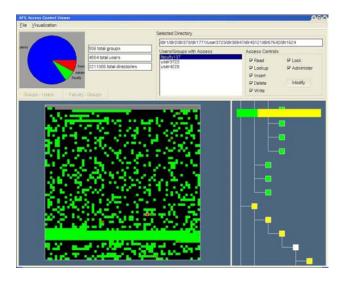

Figure 2:

In either visualization, clicking on a user brings up the user's home directory tree (lower right, Fig. 1 in an Internet Explorer-

style view. Each directory is represented by a colored icon. The color represents a warning level for that particular directory, which can be one of three values; green, yellow, or red. Currently the visualization is configured to check for users besides the owner having access to the directory, or groups having access to the directory. A green directory satisfies none of these conditions; a yellow directory satisfies one; a red directory satisfies both. A horizontal colored bar at the top of the visualization gives a rough overview of the percentages of green, yellow, and red directories in the selected tree. This visualization supports panning and zooming interaction and scrolling through the tree using the mouse-wheel.

Individual directories from the tree can be selected for more detailed information. When a directory is selected, its full pathname and a list of users and groups with access to that directory are shown. Any of the users or groups can be selected to reveal the particular access control set relating that group or user to the selected directory. This access control set can be modified through the UI; in the future these modifications will trigger the appropriate AFS commands to modify the system in real-time.

3 IMPLEMENTATION

We are developing this tool in C++ under Linux[2]. All of the drawing is done in OpenGL, using the FLTK toolkit[5] for the user interface.

4 CONCLUSION

We have presented an interactive visualization tool that facilitates exploration, navigation and management of a large AFS filesystem. By linking different types of visualizations with multiple levels of detail, the viewer supports drill-down functionality. For example, a group may be selected, then a user from that group, then a directory belonging to that user, then another group with access to that directory to reveal the individual access controls granted to that group.

We are currently enhancing the tool in a number of ways; for instance, the pixel visualization can result in very small blocks for larger systems, thus a focus+context mechanism is necessary, so that it is convenient to pick users of interest. The directory structure view also requires more scalability.

Future work on this project will entail creation of typical scenarios from a security standpoint, such as looking for anomalies, generation of alerts. In the immediate term, this will be accomplished by a simulator that will communicate with the visualization system to send changes to the underlying file system, permitting the visualization system to be studied for real-time performance, scalability, etc.

REFERENCES

[1] AFS file system. http://www.openafs.org.

[2] J. van Wijk F. van Ham. Beam trees: Compact visualization of large hierarchies. In *Proceedings of IEEE Information Visualization 2002, Oct. 19-24, Seattle, WA.*, pages 93–100. IEEE Computer Society, 2003.

[3] B. Shneiderman. Tree visualization with tree-maps:2-d space filling approach. *tog*, 11(1):92–99, 1992.

[4] B. Shneiderman, M. Wattenberg, and D. Jones. Ordered treemap layouts. In *Proceedings of IEEE Information Visualization 2001, Oct. 22-23, San Diego, CA.*, pages 73–78. IEEE Computer Society, 2001.

[5] B. Spitzak. The fast light toolkit. http://www.fltk.org.

[6] J.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.

[2]As we use public domain tools, this application can be easily ported to Windows based PCs.