

The Translucent File Service

David Hendricks

Sun Microsystems

ABSTRACT

The Translucent File Service (TFS) is a Sun Operating System (SunOS) filesystem with copy-on-write semantics. The TFS allows users both to share a file hierarchy and to have a private hierarchy into which files from the shared hierarchy are copied as they are modified. Consequently, users are isolated from each other's changes, as files in the shared hierarchy are guaranteed not to change. Files are only copied when they are modified, conserving disk space. The TFS was built to support Sun's version configuration and management tool, the Network Software Environment (NSE).

The TFS requires no modifications to existing programs to use it. The TFS also preserves the file name space, so that a user doesn't have to connect to a funny directory to use the TFS. For example, it is possible to view the directories `/usr/src` or `/bin` through the TFS.

The TFS is currently implemented as a user-level server process; it is not part of the SunOS kernel. Even though the TFS is not a kernel-based filesystem, it shows reasonable performance. This paper describes the current implementation of the TFS and the pros and cons of this implementation. It concludes with some ideas for future enhancements, including areas where performance can be improved.

1. TFS Layers

A directory seen through the TFS appears to be a normal Unix directory but is actually a number of *layers*, where each layer is a physical directory. The layers are joined by *searchlinks*. Each layer has a searchlink which contains the directory name of the next layer and is stored in a file that the TFS hides from the user. The first layer seen through the TFS is considered to be the *front* layer, and layers found by following searchlinks from this layer are considered to be behind the front layer. Figure 1 illustrates a 2-layer TFS directory named **example**, where "Revision 2" is the front layer and "Revision 1" is the back layer. The dotted line represents the searchlink from the Revision 2 **example** directory to the Revision 1 **example** directory. A layer can be thought of as a revision of a directory; a snapshot of the directory's contents at some point in time. The newest revision of the directory is the front layer, and the rest of the layers are older revisions.

2. TFS Directory Contents

The file names seen in a TFS directory are the union of the file names seen in all the layers. If a file name exists in more than one layer, the file contents seen through the TFS will be the contents of the file in the frontmost layer. For example, in Figure 1, the TFS directory **example** contains the two files **one.c** and **two.c**. The file **one.c** exists in both layers, so the file contents that the user sees will be those of the file in the front layer, or Revision 2. On the other hand, **two.c** doesn't exist in the front layer, so the user sees its name and contents *showing through* from the back layer, Revision 1. Note that the front layer is *translucent*, or selectively transparent, because files show through from back layers unless files exist in the