Changes to Programs and User View in the Sun Networked File System

by
Russel Sandberg

The Networked File System (NFS) software development plan calls for some major changes in the Unix file system semantics for remote, shared files. The changes and possible problems associated with them are outlined below. This document is not a definitive list of all of the problems that we will encounter with the NFS, but rather a description of the problem areas that will have to be addressed.

The Tough Problems

Below is a list of the problem arising from the change to NFS which we consider hard to solve. These problems generally are caused by changes to the unix file system semantics or user view of the file system.

Super-User Permission

In a workstation environment where many people are super-users on their own machines, shared files must be protected from remote super-user access. We have decided to use a global user id space (shared /etc/passwd) for the first cut of the NFS. While this guarantees that normal user access permission will work on shared files it does not address the problem of super-user access.

Our proposed solution is to make super-user permission meaningful only on local (non-shared) files. Thus, to change remote shared files as root you must login to the server that has those files local. This is a change to the user's view of the world and should not effect programs.

There is a similar problem with set-uid-root programs. These programs will set the effective uid of the user to root, but this gives super-user permission only on local files. This means that programs like chsh, mail, uucp, news, and lpr will have permission to change local files and directories only. Lock files, spool directories, and /etc/passwd will have to be local to each machine. We can make this work using symbolic links until these programs can be replaced by network services.

Revocation and File State

The NFS design calls for stateless servers. This causes problems when the status of an open file changes. For example, Unix allows normal file operations on a file which has been removed from the file system. The NFS remote server cannot know that a file is currently open when a request is made to remove it or change its status.

This change will effect programs in two ways. First, programs which use the removed open file "feature" will have to be fixed. Among the programs that will break are: mail, and sendmail. Second, programs may get an error on a file operation because the open file descriptor has become invalid. This could happen because some portion of the path name for the open file has been removed, renamed, or had its permission changed.

We will have to check programs carefully to be sure that they check for, and handle file operation errors correctly. Most of the programs that use shared files run set-uid so they will have to be fixed or replaced anyway.

Concurrent Shared File Access

The statelessness of the NFS server causes some problems when files are shared between processes on different machines. File locking, append mode writing, and atomic writes cannot be used between processes on different machines. Programs which use these features and need to live on sepperate machines will have to be changed to use a lock service and IPC.

Another aspect of the same problem is that there is no protection against overwriting a file which is the text segment for a process being paged in. The pager can detect this and kill the process, but it is hard to make it so the write is not allowed.

Time Synchronization

Shared files will have to updated with a common idea of the time of day so that programs which check file times (like make) will work. This should not break anything unless times vary widely among machines and servers. When we implement a time service the problem goes away.

Smaller problems

The problems listed below are not as serious but will need to be looked into none the less.

Shared /tmp Naming

If we allow a shared /tmp directory we will have to be sure that all programs which create temporary files use network-wide unique names. The library routine *mktemp* can be changed to use both machine name and process id in the temporary name. Programs which do not use *mktemp* should be changed so that they do. Shell scripts which create temporary files will have to be fixed also.

Fix shells — $$ should give machine

Unix Domain Sockets

The NFS will not support Unix domain sockets on a remote file system, It may not be obvious to programs which portions of the file system are local and which are remote. Until a network naming service becomes available we will have to be sure that programs that use Unix domain sockets use only path names in their local file systems.

Shared /dev

Device nodes in NFS refer to local devices whether the node is local or remote. This is not what some people would expect, but it should not break any programs.

Df

The disk usage program df will have to be changed to do something useful in the NFS world.