N94- 33806

Virtual File System For PSDS

Tyson D. Runnels

The Boeing Company
P.O. Box 24346 MS 7F-73
Seattle, WA 98124-0346
Office: 206-865-4128
Fax: 206-865-2982
runnels@zuben.ca.boeing.com

Abstract

This is a case study. It deals with the use of a "virtual file system" (VFS) for Boeing's UNIX-based Product Standards Data System (PSDS). One of the objectives of PSDS is to store digital standards documents. The file-storage requirements are that the files must be rapidly accessible, stored for long periods of time - as though they were paper, protected from disaster, and accumulate to about 80 billion characters (80 gigabytes). This volume of data will be approached in the first two years of the project's operation. The approach chosen is to install an hierarchical file migration system using optical disk cartridges. Files are migrated from high-performance media to lower performance optical media based on a least-frequently-used algorithm. The optical media are less expensive per-character-stored and are removable. Vital statistics about the removable optical disk cartridges are maintained in a database. The assembly of hardware and software acts as a single virtual file system transparent to the PSDS user. The files are copied to "backup-and-recover" media whose vital statistical are also stored in the database. Seventeen months into operation, PSDS is storing 49 gigabytes. A number of operational and performance problems were overcome. Costs are under control. New and/or alternative uses for the VFS are being considered.

Introduction

The conceptual architecture of the Product Standards Data System (PSDS) includes large-scale file storage. The plan calls for storing 80 billion characters representing the digitization of the Boeing Company's standards documents. These documents must remain rapidly available with all revisions for the lifetime of any product built using the standards. The current documents must remain immediately available for reference and revision.

Project requirements include that the system be deployed on UNIX-based computers. The preferred UNIX-based systems had, at design time, upper limits of file storage that were significantly lower than the projected maximum. Additionally, the file-management software stored files in one single directory unless manually overridden. This limitation posed problems for fixed-capacity disk drives. Given the above requirements, a solution was sought that provided large-scale storage capacity, archival storage, disaster recovery, and flexible disk-space management. This solution is called the Virtual File System for PSDS.

Project

The project, in more detail, includes a number of components. They are illustrated in Figure 1. An acronym list is provided to decipher them. Authors, using the Authoring Workstations, create or modify the digital standards documents. The documents are stored on the Standards Authority Database platform. Each digital document consists of multiple files that, together, may be displayed or reproduced on paper as a formal corporate standard. Subsets of files are routinely downloaded to subordinate platforms. One set is named Master Local Authoritative Databases. The other is named Local Authoritative Databases. Customers of PSDS retrieve and display the standards using Retrieval Workstations. A set of platforms named Derived

Authoritative Databases store subsets of the information in a form retrievable by computer applications other than the PSDS retrieval subsystems. These other applications may, in turn, support their own form of retrieval that may or may not be strictly a reproduction of the printable standard. An example is an expert system that, when posed a question by a design engineer, reasons about the information stored in an aggregate of standards - including those stored in PSDS.

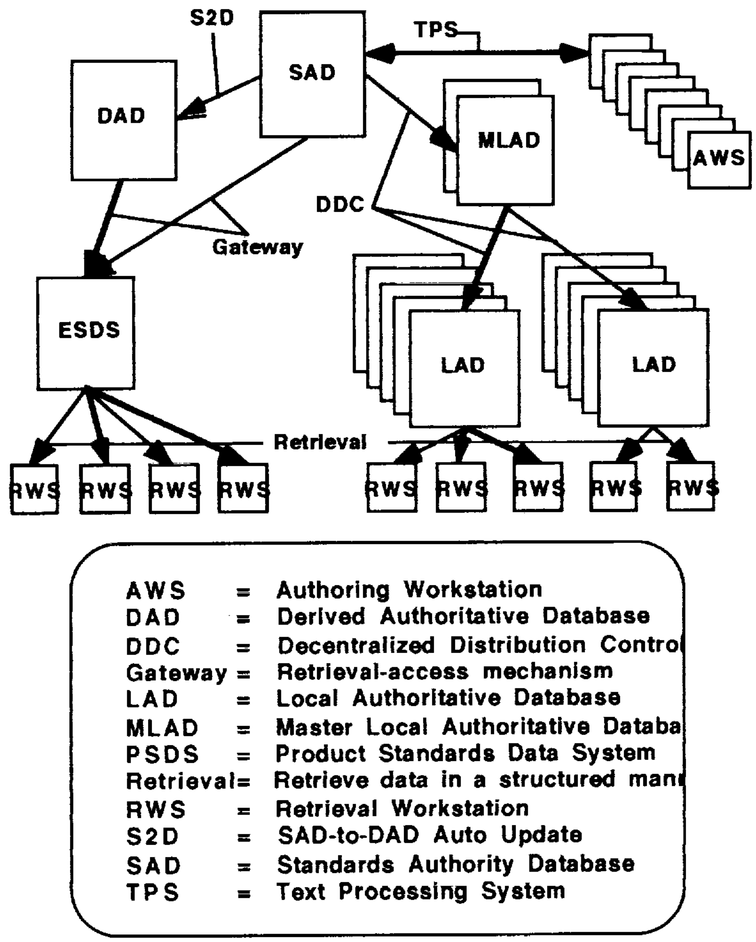

Figure 1. Global Architecture of PSDS

Objectives

The Virtual File System is a component underlying the Standards Authoritative Database platform. The objectives of the VFS include:

- Store tens of gigabytes of information (80 gigabyte projection)
- Store the information as a database and as flat files
- Support a file manager that clusters the flat files densely in a small number of directories
- Support a commercially-available database management system
- Provide "immediate" access to the information
- Behave as a permanent archive
- Secure the information through disaster-recover processes
- Be cost effective

Alternatives

The alternatives analysis was an exercise in matching cost, performance, and functionality with the objectives. Preceding decisions about architecture also constrained the choice of alternatives. A major concern for PSDS is to use technology that is available at the time of need. Although the time of need was 4 months in the future, the procurement activity alone - in a large corporation - would use 3 of them. Thus, the first decision was to use "off-the-shelf" technology.

The main UNIX server was limited, at the time, to 32 disks of no more than 1 gigabyte each. Projected storage volumes exceeded this value. Backup required 1 hour per gigabyte. A weekend would not provide enough time for a backup. UNIX-based backups also required that the applications be shutdown during backup. Backups would, therefore, exceed the shutdown time available. Additionally, the high-performance and high-capacity disk drives available for the server were relatively expensive and ill suited for archival storage.

Pure backup-and-recover software and specialized hardware did not meet the objectives either. They did not provide the required online capability. And, pure large-scale data storage products did not provide the embedded backup-and-recover functionality. The Virtual File System approach was chosen. A vendor's product met all of the objectives.

Solution

Figure 2 illustrates the interrelationships among the PSDS file-management components.

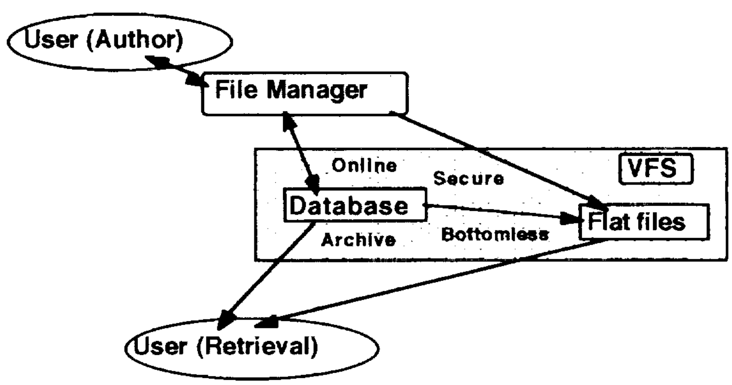

Figure 2. Interrelationship of PSDS File-Management Components

The vendor supplied the following VFS functionality:

- Hierarchical file management: The automatic migration of files from one storage media to another using a least-frequently-used algorithm.
- Embedded backup and recovery: Backups were designed to take maximum advantage of optical hardware to reduce the time necessary to perform a backup. Backup could also run while other applications were running. Recovery was optimized for disaster recovery in such a way as to reduce downtime by a factor of 10 over standard UNIX utilities.
- Lower cost-per-byte: Files are migrated to optical disk. Given careful planning, the cost per byte for data storage on the optical cartridges is less than that for the central UNIX server's spinning magnetic disks.
- Online visibility: Regardless of whether the data are on magnetic or optical, the access is transparent to the applications.
- Disk partition limits are relaxed: Disk partitions mounted on the VFS have data-storage limits extended by a least a factor of 40 over those on the central UNIX server.

Limitations and compromises are still required. First, the VFS does not handle database management systems (DBMS) that manage their own files using the disk as a "raw device" - that is, without using the UNIX file system. The PSDS database is such a DBMS. The UNIX server's disks are used by the DBMS. The PSDS file manager uses the DBMS to store the location - UNIX path name - of all of its files. These files are stored on the VFS. Second, the VFS is a separate device with its own operating system. It must be managed separately and independently yet in coordination with the central UNIX server.

Performance

Performance was estimated during the design phase to be acceptable for a network-based system such as PSDS. Actual performance was at first not as good as the estimate.

Backup and recovery are particularly slow. That is, they cannot perform their work in the time estimated to be required - or in the time available. Their performance is a function of the UNIX file structure imposed by the PSDS file manager. The file structure also slows performance of NFS and a host of other UNIX utilities and PSDS modules. The PSDS file manager's design tends strongly toward placing all files into a single directory. The UNIX file system is optimized for a tree-like structure of directories, sub-directories, and files. The VFS is optimized in the same way. Performance drops off geometrically with the number of files in a single directory. See Figure 3, next page.

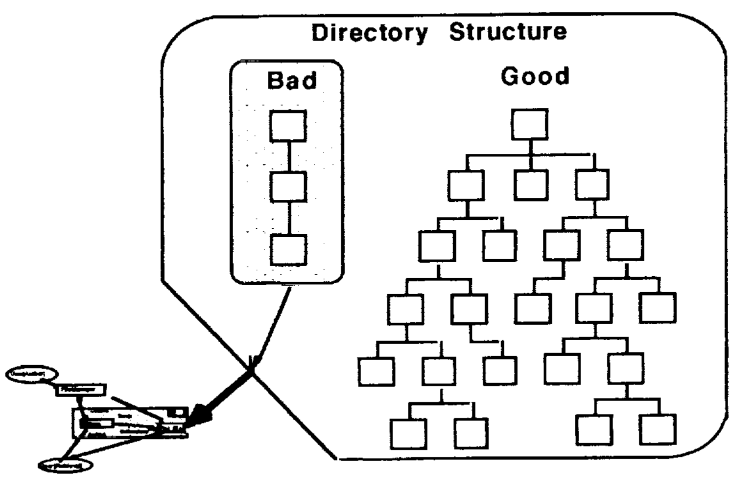

Figure 3. Major Source of Performance Problems

The Network File System (NFS) performance was also slower than expected. The NFS performance is partly a function of the speed of the central processing unit (CPU). The VFS is not a fast CPU in comparison with the main UNIX server and the PSDS load is large. This problem is overcome in the current system by using the central UNIX server's disks as a work area for the most time-critical files. This means unanticipated system management.

Futures

The VFS is being used as a bottomless archive with backup and recovery for the SAD. At least three other distinct possibilities exist for use within the capabilities supplied with the VFS. See Figure 4.

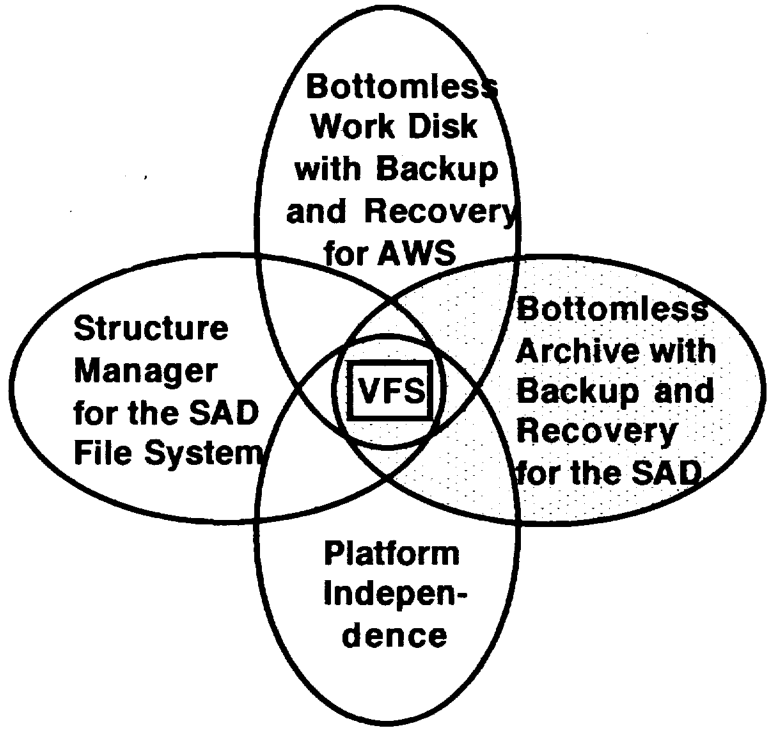

Figure 4. Potential Future Applications for a VFS

The VFS capability may be extended to components of PSDS other than the SAD. The Authoring Workstations (AWS) receive original work from authors on a daily basis. The disks on the workstations may be made into virtual file systems just as are those on the SAD. Linking the AWS to the VFS would provide the bottomless partition feature and, most importantly, a centralized backup and recovery mechanism. A drawback to this process is that both the local-area and wide-area networks will receive more traffic.

Another feature of the VFS is to manage the distribution and proliferation of sub-directories in a way transparent to the PSDS file manager. Thus the "bad" directory structure can be made into the "good" directory structure independently from the requirements of the application (the PSDS file manager). Tests on PSDS data show this to be a 100% improvement in performance for UNIX utilities, NFS, and backup-and-recovery. A drawback to this process is that the system administrators have an added burden of maintaining a mapping of files from the "bad" structure to the "good" structure.

The third additional way to use the VFS is to distribute it among the far-flung platforms that comprise PSDS. The original VFS acquired by PSDS was an independent "turnkey" system - hardware and software. Evolution of the VFS is moving it toward a more software-only architecture. Limitations of CPU speed, memory constraints, number of I/O busses, and sundry become less restrictive. Each local-area-network could have a VFS. System-administration tools are also expected to evolve in support of a more distributed architecture. Traffic on the wide-area-network could be reduced. Drawbacks are cost and training. The VFS is not trivial to manage. It is not trivial in cost.

Summary

Given its requirements and constraints, PSDS picked a solution for large-scale file storage that worked. Conversely, a solution was available that satisfied the PSDS requirements and constraints. Opportunities for wider use of the VFS exist and are being considered.

Problems were encountered after installation. They included issues involving cost, performance, and reliability. The issues were attacked vigorously by PSDS and vendor staff and resolved.

Future management of PSDS data will be supported by enhancements from the VFS vendor, improvements in the PSDS software, and improvements in UNIX-based systems. It appears, though, that the volume of data will continue to exceed the currently available "simple" storage systems and that a VFS in some form will be required.