Mirror File System

New DNA for Computer Systems

A Twin Peaks Software White Paper
April 2009

Any single resource is vulnerable to catastrophic failure.
Redundancy provides a safety net, but is that enough?

Overview

Although current technologies provide data and system redundancy—RAID disks, storage replication, data base replication, not to mention clustering or fault-tolerant systems—they are all designed to form a safety net in case a single resource becomes unavailable. This paper shows how Twin Peaks Software's Mirror File System goes beyond conventional redundancy issues to provide a new basis for IT infrastructure.

Problem

The single resource vulnerability problem is "in the DNA" of computer system design and information processing on storage devices, since nearly all computer systems are based on the single, standalone model. This includes operating systems, which manage processing, and file systems, which manage information on storage devices. Both inherit this DNA, which is why they tend to store and retrieve information on a single, standalone computer system. The computer system, operating system, and file system all operate on a single entity. That is the problem.

End User System

Every laptop or desktop user who knows that there is only one copy of a file on their system would want to back up their files to another disk or computer system just in case the computer or disk is not working. One can manually copy the files to another disk drive or another computer system. But this is a tedious and cumbersome task that most users either forget or are reluctant to do. Shouldn't this be handled automatically?

Enterprise System

Enterprise IT administrators need to provide fast, reliable, uninterrupted service to their clients around the clock.

RAID disk arrays and redundant storage try to address the vulnerability of the single disk and storage model. Even when the disk or storage subsystem is protected, however, the single system connected to it remains vulnerable, so another computer system or node can be added to form a clustering system. Redundant storage and clustering systems mitigate the single resource issue to some extent, but are not adequate to handle disaster scenarios, for instance when fire or earthquake causes the whole redundant setup to shut down. They are also expensive and inefficient in terms of resource utilization.

Solutions

Today's file systems, such as EXT3, NTFS, NFS, and DFS, can only manage information on a single system, whether local or remote (Figures 1 and 2).

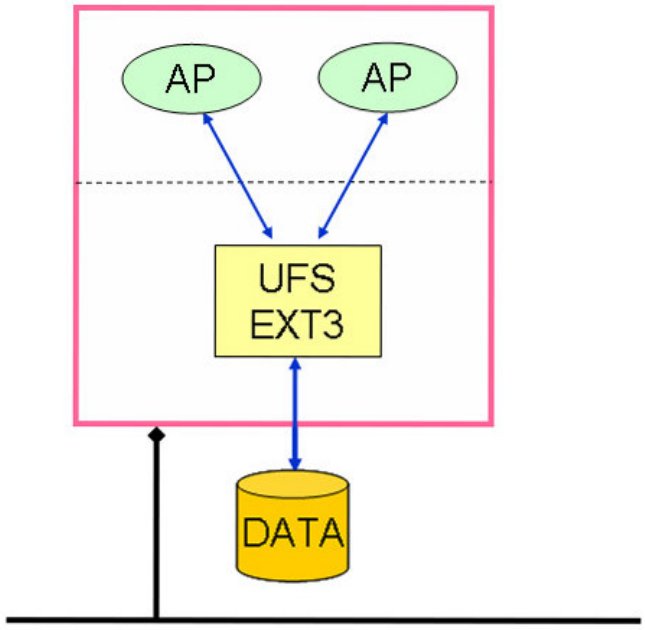

Figure 1: Local File System

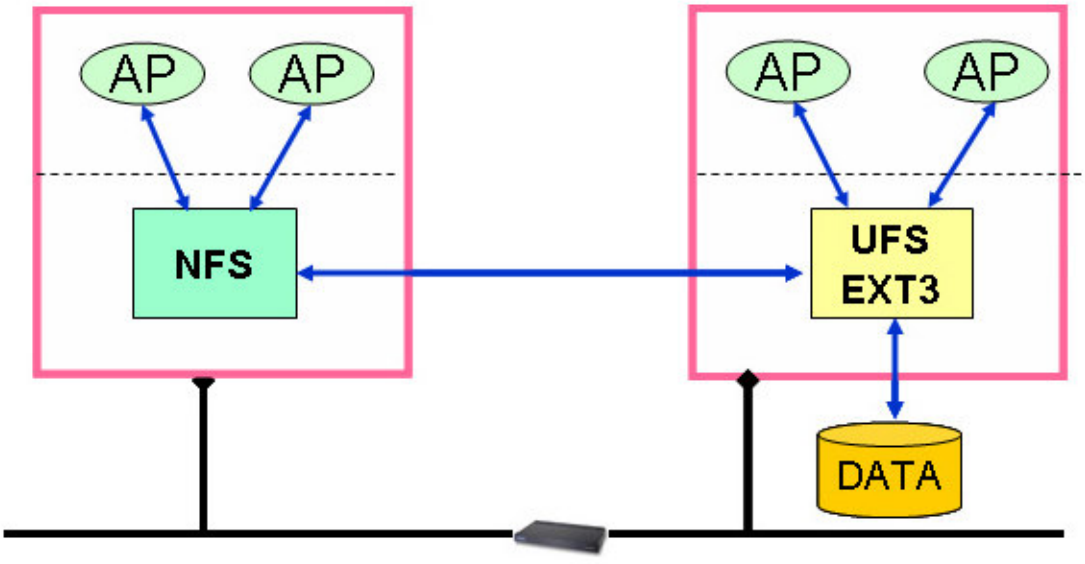

Figure 2: Network File System

For information to exist on multiple systems, it must be copied either manually or on a schedule (cron jobs). For many users and enterprises, this approach is the best available today and has often been considered a "good enough" solution. The Mirror File System puts information on multiple systems in real time. This is a better — in fact, the best — solution to today's redundancy problem.

As shown in Figure 1 and 2, EXT3 or NTFS file systems manage files on a local system; NFS or CIFS file systems manage files on remote servers. A user or application can use either one, but not both simultaneously (Figure 3).

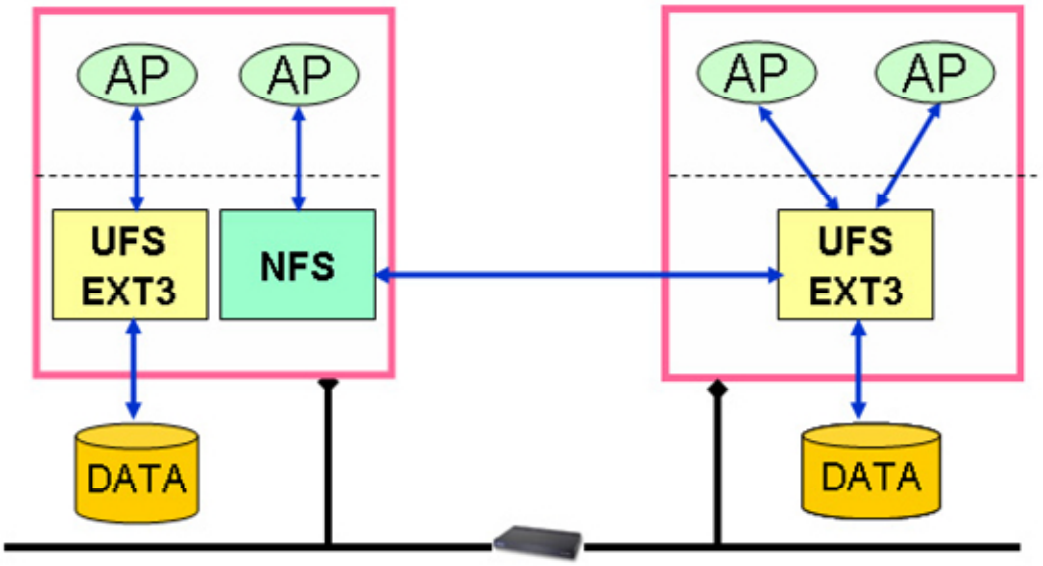

Figure 3: Local File System OR Network File System

The Mirror File System (MFS) resides on top of EXT3 and NFS, in the kernel space, and manages EXT3 and NFS (Figure 4). All file operations from the application go first to the MFS, which receives the file operation and forwards it to EXT3 for READ-related operations and to both EXT3 and NFS for WRITE-related operations. When a user creates a new file or updates an existing file, an identical copy is created or updated on both systems in real time. Both (or multiple) instances of same file are live and can be used immediately.

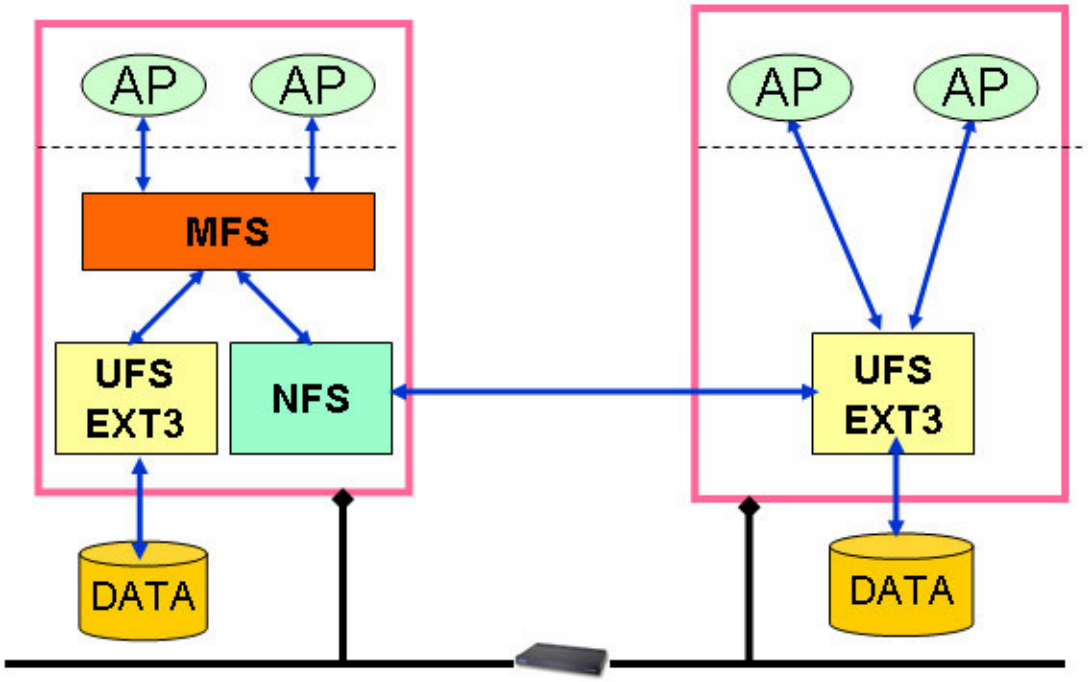

Figure 4: Mirror File System = Local File System + Network File System

MFS Usage

End User

Whenever a user creates or updates a file, it is instantly updated on both systems — for instance, on the user's laptop and on a remote server — in real time, and both copies can be viewed immediately. This solves the file backup problem and also provides a new way of sharing information. In fact, this amounts to a cloud computing solution for file backup.

Enterprise System

Whenever a file on one server is updated, the changes are replicated to another remote server in real time. There are always two identical copies of any given file on the network. Replication can be bi-directional, with both servers able to write to the same file at the same time while maintaining file consistency. Even, or perhaps especially, if the two servers are deployed in geographically dispersed locations, identical files remain available and accessible wherever needed.

Advantages of MFS File Backup

Conventional online backup falls into two categories: Old fashioned manual or scheduled backup and file system filtering. These approaches are discussed briefly below and contrasted with the MFS approach.

Manual and Scheduled Backup Approach

An application starts to scan a directory or folder either manually or at a specified interval, such as hourly or daily. When it finds a file that has been changed since the last scan, it backs it up to the remote server.

There are two problems with this approach. One is that even if there are only small changes to a large file, the application still backs up the entire file to the remote server, wasting CPU resources and network bandwidth. The application is not capable of backing up only the changed portion of a large file to the remote server. The second problem is that when the folder contains a large number of files, the application has to scan all of them to find out which one has been changed. This too consumes system resources and network bandwidth.

File System Filter Approach

A file system filter module in the kernel intercepts and saves file operations and data from the application to an internal buffer. An application reads the operations and data from the internal buffer and sends them to the remote server, where another application receives the data and replays the operations and data to the local file system. This is a better approach than manual or scheduled backup, but it creates additional technical complexity. In order to perform the replication, it requires a file system filter and the cooperation of ad hoc applications on both the local system and the remote server. Under heavy traffic conditions, the internal buffer tends to overflow, preventing the local ad hoc application from reading data from the buffer, or writing to the remote ad hoc application, fast enough.

MFS Approach

The MFS is a kernel file system module. It receives file operations and data from the application, then forwards them to the underlying file system: EXT3 or NTFS for the local copy, NFS or CIFS for the remote copy. The underlying file systems then process the file operation and data normally. (They are not aware that the operation and data come from the MFS.) Since the file systems know where to store new data, only the changed portion of a file needs to be sent, not the entire file. This building block approach utilizes existing file system infrastructure to perform file mirroring in real time. It is elegant and efficient and does not need ad hoc applications or a new file transfer protocol.

Real-Time File Mirroring vs. Conventional Block Device Mirroring

Block device replication is implemented as a device driver in the host system or in detached hardware storage subsystem (Figure 5). The block storage device consists of a pair of identical devices connected by a channel. When the device driver receives data from the upper module in the kernel, it writes to the local device and sends the same data to the remote backup device in real time.

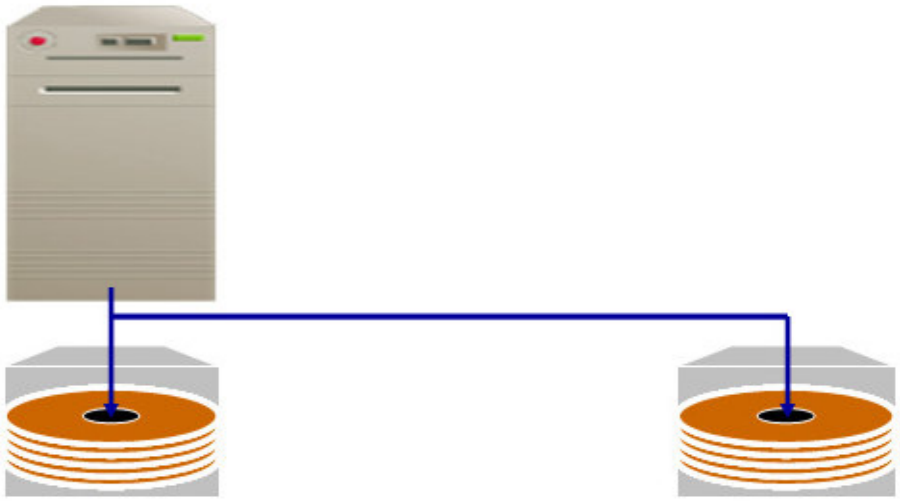

Figure 5: Disk storage can be replicated either locally or remotely

There are always identical copies of data on both devices; however, there are also two big problems:

- Sending the data from the local device driver to the remote backup device driver requires a new data transfer protocol.
- Because the replicated data on the remote backup device is updated through the device driver, rather than through the file system module, it cannot be accessed right away.

The backup device can be used only when the primary device goes down. When that occurs, either the local host or the remote backup host can mount and use the backup device. Before it mounts a device, the host system must check (fsck) the data for corruption and inconsistency and repair it as necessary. Corruption and inconsistency can occur when data is replicated from the primary storage device, usually due to abrupt shutdown of the primary system or transmission errors during replication.

In other words, block device mirroring requires two storage devices, but uses only one, treating the other one as a spare. There is no guarantee that the data on the spare, or secondary, or backup device will be good, valid, and available when the primary system fails.

MFS uses the underlying EXT3/NTFS to write data on the local device and NFS/CIFS to write the same data to the remote system. All of the underlying file systems are stable; NFS/CIFS is the industry standard file transport protocol. All file operations and data go through a sanity check to make sure they do not corrupt the physical file system on the storage device. The same file is live, not a raw disk block, on both systems, and can be viewed and used right away because it is already mounted. The replicated data file can easily be verified by looking at it when it comes in. When one device or system goes down, another stays up and running to take over services right away.

Geographically Dispersed Clustering vs. Conventional Clustering

Clustering uses two systems, a primary and a secondary, both connected to a shared disk (Figure 6). But only the primary node can access the disk at any time. The secondary node is also physically connected to the shared disk, but it can access the disk only when the primary node is down.

Conventional cluster systems have many problems. For instance:

- The disk is single resource for both nodes. If it is down, neither node can access the disk.
- Using two systems to perform one task is not good resource utilization.
- The two nodes have to be physically close to the disk, probably in the same lab or similar location.

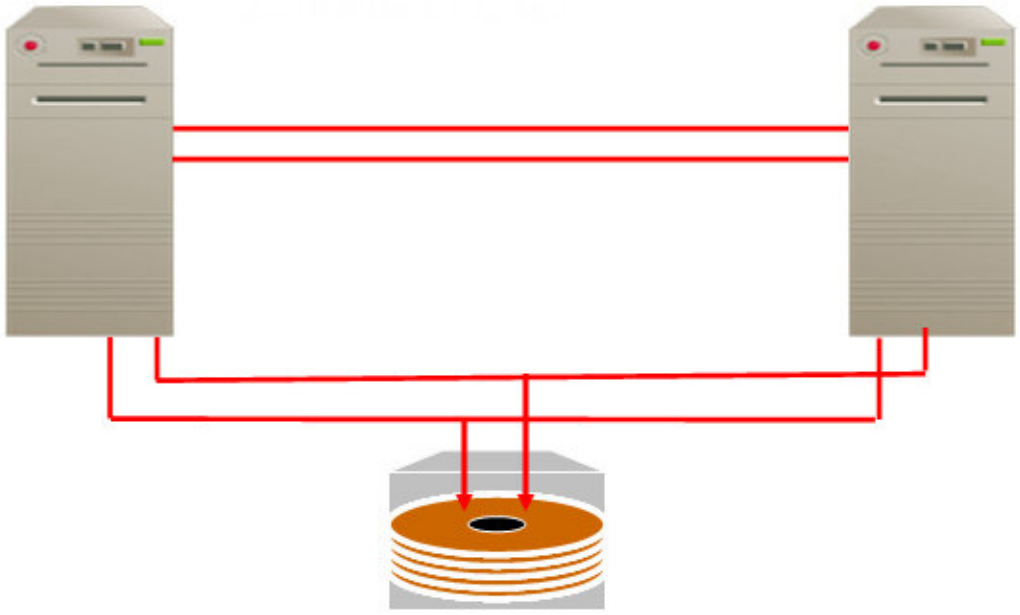

Figure 6: A two-node clustering system operates like one node on a single location

MFS can turn a single-location cluster system into a geographically dispersed cluster system in which each node connects to its own disk (Figure 7). Any update to a file on one node also updates the same file on the other node in real time. Both nodes remain up and running to provide the same services to their clients although the two nodes can be separated by hundreds or thousands of miles.

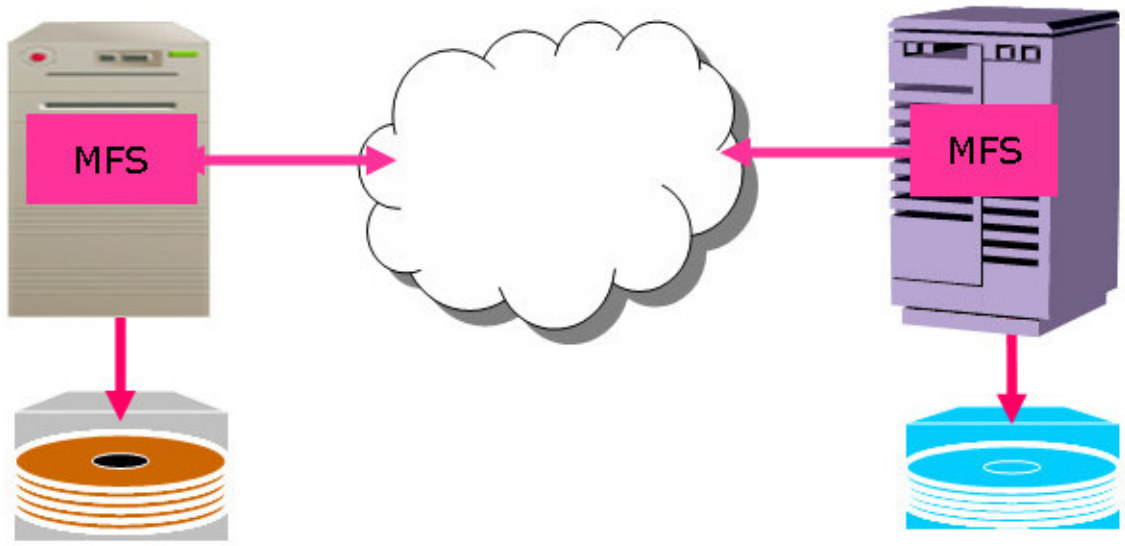

Figure 7: With MFS, second systems can be located anywhere in the Cloud

Benefits

Here is a brief summary of MFS benefits for IT.

Disaster Prevention, not Disaster Recover

When disaster strikes, it is already too late to recover lost time and information. MFS makes sure that an up-to-date copy of any file remains available to *prevent* disaster instead of recovering from it.

Always Available, not Highly Available

Conventional high availability clustering systems are struggling with how to define *highly*, for instance, by many "9's" of availability can be achieved (90%, 99%, 99.9%, etc.), approaching availability as a limit. MFS transcends this problem by making data *Always Available* to users and applications.

Load Balancing Geographically Dispersed Servers

The advantages of load balancing two or more servers are well known. Utilizing load balancing on geographically dispersed servers enables faster delivery of services, with redundant copies of data remaining available.

Flexibility in Choice of Computer and Storage Systems

Conventional clustering systems and disk/storage-based replication products all require that the same type of computer system and disk/storage be used as backup devices. MFS has no such restrictions and can utilize existing heterogeneous infrastructure investment.

Real-time Online File Backup

Users can perform online file backup easily and transparently, and users can share their online file with others.

Enhanced Cloud Computing

The fact that identical copies of a given file exist on the users' desktop and in the Cloud enables users to do conventional desktop computing as well as Cloud computing.

Conclusion

Early mainframes and minicomputers were limited to reading and writing files from a local storage unit attached to individual, standalone systems. In the 1980's, NFS enabled standalone systems to read and write files on a remote server as though they resided on the local system. This was a significant advance, but it still was not able to address the vulnerability problem of the single resource model.

The Mirror File System, the first file system that can write the same file to two or more different systems in real time, breaks through the barrier that formerly confined computer system to the single, standalone resource model and enables single systems to be integrated into the Cloud or geographically dispersed network cluster systems. Making identical files available to every system on a given network also provides substantial benefits for disaster recovery and online file backup.

In the near future, we expect to see data center servers in widely dispersed locations all be able to contain the same data, constantly updating one another in real time. For users, this means their files can be seamlessly backed up every time a file on their desktop or laptop is

changed and saved. For enterprises, it can prevent the need for disaster recovery.

MFS opens up new ways of deploying information on the network that were previously thought impossible. Its design concept breaks through barriers that have historically restricted computer systems to be administered as single, standalone entities, and it enables a single computer to manage multiple, identical resources in real time on the network. In this sense, it provides new DNA for computer design and offers the possibility of a new landscape for information technology.