Topics in SFS Internals [10]

The data locking capabilities of the NC1FS file system are not always sufficient to allow the access control that is necessary when multiple processes on different Cray Research systems require the same degree of controlled, shared access to a file. For this reason, SFS file systems support the following extensions to the NC1FS file system which make the sharing of file systems between cooperating Cray Research systems possible:

- File system meta-data cache coherency
- Mandatory locking
- Lock ownership by system
- Media (sector) update protection

These features are described in the following sections.

10.1 File system meta-data cache coherency

Traditional (non-shared) high-performance file systems, including the NC1FS file systems, derive some degree of their high performance by keeping often-referenced data structures in cache memory. By avoiding the overhead of reading and writing often-used file system data structures from and to the file system media, significant performance gains can be realized.

The problem that this technique poses for SFS file systems is that only one Cray Research system in an SFS cluster can read the cached data: the Cray Research system whose main memory is acting as the cache. If the cached data is ever modified, as it will be if the file is deleted or changes size, the rest of the systems in the cluster can only read old information (the file system information that resides on-media), which is out-of-date with respect to the cached information visible only to the Cray Research system that owns the cache.

The UNICOS SFS feature solves this problem using the straightforward approach of keeping file system data up-to-date on the file system media all the time. No cacheing of file system information by any of the Cray Research systems sharing a file system is allowed, with one notable exception: if a file is locked, the UNICOS system that owns the file lock may cache file system information related to the locked file after the file lock has been recorded on the file system media so that the other Cray Research systems sharing that file system can determine that the file is locked.

10.2 Mandatory locking

In the NC1FS file system, data locking is considered discretionary; that is, it only works if the programs that will be using a shared file cooperate and agree on how they will utilize and react to data locks. With the UNICOS SFS feature, data locking is automatically mandatory in nature; that is, the synchronization of contending programs is handled automatically by the UNICOS operating system.

10.3 Lock ownership by system

A lock on a file is said to be owned by a particular mainframe in an SFS cluster. The concept of file locks being owned by computer systems, and not simply by processes running on those systems, is the second key extension to the NC1FS file system that makes the sharing of file systems possible.

When a file is locked on an NC1FS file system, or on another non-shared file system that supports file locking, some information about which process on the system locked the file is recorded as part of the lock. This information is used for purposes such as making sure that only the process that locked a file may unlock it.

On NC1FS file systems, no information about which computer system the process that locked the file is running on is recorded; this information is not needed because it is assumed that only one computer system can be accessing the file system. In a UNICOS SFS environment, this single-system assumption is no longer valid, so the concept of lock ownership by systems as well as processes has been introduced.

From a shared file system point of view, "which system has this file locked?" is at least as important a question as "which process has this file locked?" Changes have been made to the NC1FS file system inode data structure to be able to record lock ownership by UNICOS systems as well as by processes to support this concept.

10.4 Media (sector) update protection

NC1FS file system inodes are stored on file system media with many inodes packed into a single media sector, where a sector is the smallest unit of data that may be transferred to or from the media. Typically, on a network disk, several hundred inodes are stored in a physical sector of the media holding the file system.

Separate and distinct inodes, whose only relationship may be that they happen to reside in the same physical media sector, may be independently accessed and updated in a non-shared environment because the media sectors holding these inodes are only being accessed by a single system.

In a shared environment, updates to inodes must involve an extra locking operation, which protects the sector containing the inode being updated from simultaneous updates from other systems.