NFS gets revved in Solaris 2.5

Summary
While it may have earned widespread use, NFS still has room for improvement. With the release of Solaris 2.5, Sun will deliver a new version of NFS and will let NFS run over TCP. Customers can expect enhanced network performance throughout, and new features that improve flexibility, usability, performance, security, and robustness. Let's take a look at the key NFS-related features and how they'll impact your site.

By Brian Wong

The advent of Solaris 2.5 promises many enhancements to Sun's flagship OS, including an improved implementation of the NFS protocol. Two items in the new release will have significant impact on networks: NFS version 3, and the ability to run NFS (either version) over connection-oriented protocols such as TCP. Of these two, the most significant new addition is the NFS version 3 protocol offering.

NFS V3 is related to the now-universal version 2 protocol, but some significant changes enhance its performance. These changes include:

- A safe asynchronous-write protocol that accelerates NFS writes
- Finer access control
- Less overhead
- Bigger file-transfer sizes
- Support of NFS over TCP

Still safe, but faster

One of the least-endearing features of NFS version 2 is the very-safe-but-very-slow nature of its fully synchronous write operations. The completely stateless nature of the version 2 protocol requires that the server commit write operations to stable storage before acknowledging them. Since most Unix users are accustomed to lazy write-behind buffering for local disk writes, NFS V2 writes seem sluggish.

Several products accelerate writes, including Legato's PrestoServe and its Sun relative, a battery-backed SIMM. Various programmatic approaches have also appeared, including write clustering, which is found in Solaris 2.x and Interstream's EFS.

Although effective, these workarounds address the symptom rather than the cause of NFS V2's slow performance. (Some vendors even offer unsafe asynchronous writes in an attempt to boost throughput.) The new protocol allows for clustered asynchronous writes combined with a safe two-phase commit protocol that permits clients to stream to the server at far higher rates than possible under the older protocol.

Fine-grained access control

Another important extension to the protocol is the provision for much finer control of file and directory access. The version 2 protocol does not provide clients with sufficient information to accurately determine the actual permissions available on a file system object. (The best known example is the

mapping from "root" to "nobody".) Although this has served well for many years, users are beginning to require finer-grained access control.

The new protocol provides a mechanism for clients to submit credentials and request access checks that exceed the limited semantic definitions of the older protocol. This permits the client to ask the server to verify the available permissions. Access control lists (ACLs) are not explicitly defined in the NFS protocol, so Solaris 2.5 implements a private protocol that lets clients obtain and edit ACLs on remote file systems. When combined with the NFS version 3 access mechanism, this permits access to be controlled at an extremely fine level. (A private protocol is required because there is as yet no industry-standard definition for ACLs. The private protocol will be replaced when such a standard is defined.)

Less overhead

A number of other improvements were made to increase over-the-wire efficiency. For example, one of the most frequently used version 2 operations is `getattr`, which obtains file attributes (such as name, permission mask, last modified time, etc.) for a referenced file. Studies of existing NFS sites reveal that virtually all `getattr` operations are proceeded by an acknowledgment for the previous operation, and that the overhead of processing two discrete operations is substantial. Accordingly, the V3 protocol has been modified to permit file attributes to be returned "piggybacked" on the acknowledgments for other operations. This improves the overall efficiency of the protocol stack by eliminating extra packets and extra NFS operations.

Larger file-transfer sizes

In a similar vein, the transfer size maximum is raised from 8 kilobytes in version 3 to as much as 4 gigabytes. This permits clients and servers to exchange data in much larger units for greater efficiency. The Solaris 2.5 implementation negotiates for block sizes as large as 64 kilobytes, resulting in point-to-point throughput as high as 5.4 megabytes/second on current SuperSPARC-based platforms (which slightly outpace even hyperSPARC-based computers on this code). The 64-kilobyte size is a result of the use of standard TCP and UDP transport protocols, which have a 64-kilobyte maximum window size. A future release of Solaris will implement RFC 1323, which provides for larger windows when operating over TCP.

Sixty-four-bit file offsets in NFS 3
let suitably-equipped clients and servers
manipulate files of essentially arbitrary size.

Another major NFS improvement is the transition from 32-bit file offsets to 64-bit file offsets in the over-the-wire packets. This permits suitably-equipped clients and servers to manipulate files of essentially arbitrary size. There is no practical limit on the size of a filesystem that can be manipulated with either protocol. The distinction between the protocol and the implementation is crucial in this area: The version 3 protocol uses 64-bit offsets. (Note that this is entirely possible on existing 32-bit hardware.) The Solaris 2.5 implementation at present does not make use of this ability since the underlying Unix filesystem and Solaris virtual memory systems are not fully 64-bit clean, but a future release of Solaris will offer large local files. Solaris will support large files over NFS at that time.

NFS over TCP

The other major innovation in the Solaris 2.5 NFS implementation is the ability to run the NFS application-level protocol over TCP. Because TCP is a reliable transport protocol with congestion control and built-in error handling, it provides a simpler mechanism for the NFS protocol suite than the more traditional UDP transport. This is a minor enhancement in the common LAN environment, but

for those wishing to implement wide-area file sharing, NFS over TCP makes for smoother transmissions -- even when operating over relatively noisy and error-prone communication lines.

As with the NFS V3 and V2 protocol selection, the use of TCP or UDP transport is easily selected at mount time. Either transport is available for either NFS protocol; the default is to run NFS version 3 over TCP. This is the selection made by Solaris 2.5 clients, but Solaris 2.5 servers will respond immediately to any of these combinations in order to support heterogeneous environments.

Backward compatibility

Although the NFS version 3 protocol behaves very much like its predecessor from a user's perspective, the substantial nature of the changes make version 3 a completely different protocol. Version 2 clients cannot communicate with version 3 servers, and vice versa. Solaris 2.5 avoids this problem by offering simultaneous V2 and V3 servers and clients; V3 is the default protocol requested by Solaris 2.5 clients, but if no V3 server responds to the mount request, the clients will fall back and attempt V2. The servers likewise offer both V2 and V3 services, making full interoperability possible. (The `nfsstat -m` command can be used on the client to discover what protocols are in use.)

One of the consequences of the divergence of the version 2 and version 3 protocols is that the industry-standard SPEC_sfs.097 NFS file server benchmark (popularly known as LADDIS) must be substantially revised to account for the peculiarities of the new protocol. The SPEC committee is working to revise the benchmark to provide metrics for the new protocol.

It's worth it

The Solaris 2.5 NFS implementation provides the biggest enhancement to NFS functionality since Solaris 1. Improvements have been made in the areas of performance, efficiency, flexibility, security -- and, when combined with Solstice HA-NFS, improved reliability and availability.