

Reiko Kaps

Daten-Schnellstraße

Netzwerkplatten mit NFSv4

Die Version 4 des Network File System (NFSv4) räumt das in die Jahre gekommene Dateisystem gründlich auf. Sie vereinfacht die Einrichtung, verbessert die Kommunikation zwischen Server und Client und sichert optional die Übertragung sowie die Anmeldung.



Das unter Unixen wie Solaris, den BSD-Abkömmlingen und Linux verbreitete Network File System gilt als sehr schnelles Netzwerkdateisystem, dessen Version 3 jedoch ins Alter gekommen ist. Es legt viele Mängel an den Tag, die sich aus der langen NFS-Entwicklungsgeschichte erklären: So galten Mitte der 80er Jahre lokale Netze noch als sicher, denn sie waren meist nicht an Weitverkehrsnetze angebunden. Verschlüsselung spielte in LANs damals kaum eine Rolle und als Ausweis reichte die IP-Adresse.

Mit NFS lesen, schreiben, löschen und erzeugen Rechner Dateien direkt im Dateisystem des Servers. Anders als bei FTP muss NFS diese Dateien nicht erst auf den lokalen Rechner übertragen, wenn ein Programm in eine Datei schreiben will. Es zählt daher zu den verteilten Dateisystemen.

Anwendungen wie beispielsweise ein Texteditor öffnen und lesen Dateien über Systemaufrufe (open, write, mkdir ...), die NFS als Netzwerkpakete an den Server sendet. Um diese Befehle über Betriebssystemgrenzen hin-

weg auszutauschen, setzt NFS die Daten mittels der External Data Representation (XPR) in ein plattformunabhängiges Format um und sendet sie als Remote Procedure Calls (RPC), also als Funktionsaufrufe, an den Server.

Entwickelt wurde NFS Mitte der 80er Jahre durch Sun, die das Dateisystem mit Sun OS 2 auf den Markt brachten. Mit dem Erscheinen von Sun OS 2.5 stellte der Hersteller NFSv3 vor, das unter anderem deutlich flotter als der Vorgänger ist und die Begrenzung der Dateigröße auf zwei Gigabyte abschafft. Sun hat

Ende der 90er Jahre die Entwicklung von NFS an die IETF (Internet Engineering Task Force) abgegeben. Ein erster Entwurf für den NFSv3-Nachfolger stammt aus dem Jahr 2000. 2003 folgte RFC 3530, das NFSv4 als offenen Internetstandard definiert [1]. Die NFSv4-Entwickler zielen besonders auf die Sicherheit und die Geschwindigkeit des Protokolls. Außerdem hat die IETF bereits die NFS-Version 4.1 vorgeschlagen, die das Dateisystem um Sitzungen, Verzeichnisdelegation und paralleles NFS erweitert [2].

NFSv3 mangelt es besonders an einer zeitgemäßen Authentifizierung, mit der sich beispielsweise Benutzer sicher am Server anmelden können. Stattdessen weisen sich die NFS-Clients nur über ihre IP-Adresse oder ihren Hostnamen aus, die Angreifer leicht fälschen können. Benutzer authentifiziert das Protokoll überhaupt nicht. Die Zugriffsrechte auf die freigegebenen Dateien regelt NFS über die Benutzer- und Gruppenkennungen des Servers, die die Clients als numerische Kennung übertragen. Verschlüsselung war zwar bei NFSv3 bereits über Secure-RPC möglich, doch stand diese Technik nicht auf allen Betriebssystemen bereit und fand daher wenig Verbreitung.

Umbauten

NFSv4 authentifiziert zwar immer noch keine Benutzer, wie das etwa der CIFS-Server Samba macht. NFSv4-Clients und -Server können sich jedoch über Kerberos 5 sicher ausweisen, das sich auch zur Verschlüsselung eignet.

Kerberos wird seit Ende der 1970er Jahre ständig weiterentwickelt und liegt mittlerweile in der Version 5 vor [3]. Die Sicherheits-Infrastruktur besteht aus einer Reihe von Diensten (Admin-Server, Distribution Center, Kerberos-Clients). Sie eignet sich besonders für lokale Netze und authentifiziert nicht nur einen Benutzer oder Client bei einem Server, sondern sorgt auch dafür, dass der Server sich umgekehrt beim Client ausweist. Damit Server und Client keinem Hochstapler aufsitzen, verbürgt der Kerberos-Dienst auf ähnliche Weise bei beiden seine Identität. Man-in-the-Middle-Angriffe lassen sich damit sicher unterbinden, allerdings benötigt das Verfahren sehr viel Aufwand beim Einrichten. Neben der Authentifizierung sorgt Kerberos auch für die nötige Verschlüsselung des NFS-Datentransfers. NFSv4 verbindet sich über RPCSEC_GSS-Authentifizierung respektive über das Generic Security Service Application Program Interface (GSS-API, RFC 2743) mit Kerberos 5.

NFSv3-Server und -Client wissen nur wenig über ihre Aktionen wie Lesen, Schreiben und das Setzen von Dateisperren, sodass bei falscher Einrichtung schnell einmal Daten verloren

gehen können oder sich das Protokoll bei den Dateisperren verhaspelt. Die Folge sind beispielsweise nicht startende Programme oder Abstürze, die sich nur schwer erklären lassen. Trotz aller Zusätze wie lockd ist eine NFSv3-Verbindung zustandslos. Erst unter NFSv4 melden sich Clients beim Server an und erhalten von ihm eine Client-ID, die der Server nach einer gewissen Zeit (lease time) verwirft, falls der Client sie nicht regelmäßig auffrischt. Öffnet ein NFSv4-Client eine Datei auf dem Server, erhält diese eine Zustands-ID (Stateid), die Auskunft über ihre aktuelle Nutzung gibt.

Übertrug NFS früher jeden Systemaufruf einzeln, bündelt NFSv4 nun bestimmte Befehlsfolgen in einer einzigen Anfrage. Will beispielsweise ein NFSv4-Client aus einer Datei lesen, so verbindet er die Befehle Lookup, Open und Read in einem Compound Remote Procedure Call, was Netzwerkverkehr spart. Diesen Kombi-RPC arbeitet der Server der Reihe nach ab – bei Fehlern bricht die Verarbeitung einfach ab und der Server informiert den Client.

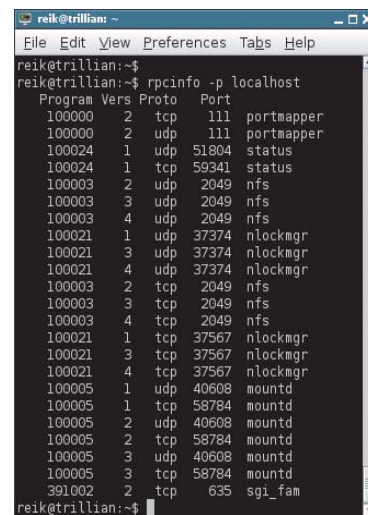
Ein NFSv4-Server kann Dateioperationen an einen Client delegieren. Ein Client verändert oder löscht eine delegierte Datei damit im eigenen Cache und spart wiederum Netzwerkverkehr. Erkennt der Server, dass andere Clients auf die Datei zugreifen wollen, widerruft er die Datei-Delegation und der Client schickt seine Änderungen zum Server. Solche Rücknahmen bewerkstelligt NFSv4 über Callback-RPCs. Da Firewalls die Rückrufe blockieren

könnten, testen NFSv4-Server beim Verbindungsaufbau die Fähigkeiten der Clients und passen ihr Verhalten dem Ergebnis an.

NFSv4 vereinheitlicht die Interpretation und die Anwendung von (erweiterten) Dateisystem-Zugriffsrechten (ACLs) zwischen POSIX-Betriebssystemen wie Solaris, BSD oder Linux auf der einen und Windows auf der anderen Seite. Es beherrscht benannte Dateiattribute, in denen Anwendungen eigene Angaben hinterlegen können.

Die Benutzer- und Gruppeninformationen überträgt NFSv4 als Klartext, nicht mehr als numerische Kennung. NFSv4 setzt Kennungen und die Namen der Benutzer und Gruppen mittels des idmap-Dienstes um. Die Inhalte dieser Zugriffs Kennungen kodiert es in UTF-8 und kennt damit auch exotische Schriftzeichen. Zusätzlich verbessern diese Neuerungen die Zusammenarbeit mit anderen Netzwerkdateisystemen wie CIFS unter Windows oder Samba.

NFSv3 besteht aus mehreren Einzelprotokollen, die jeweils Teile der Kommunikation zwischen Server und Client regeln: Das Mount-Protokoll steuert den Zugriff auf die Freigaben, Lockd setzt Dateisperren, die Statd beim Server und anderen Clients registriert. Die Teilprotokolle benötigen für ihre Remote Procedure Calls zudem einen Portmapper-Dienst (portmap), der den RPC-Diensten Portnummern dynamisch zuteilt. Will oder muss man NFSv3 durch eine Firewall oder einen SSH-Tunnel nutzen, muss man neben dem eigentlichen NFS-Port 2049 auch den Port-



Unter NFSv3 teilt der Portmapper den Teilprotokollen Ports dynamisch zu.

mapper-Port (111) und die Ports für mountd, lockd und statd weiterleiten, die man in diesem Falle per Hand vorgibt. Sonst wechseln sie bei jedem Start.

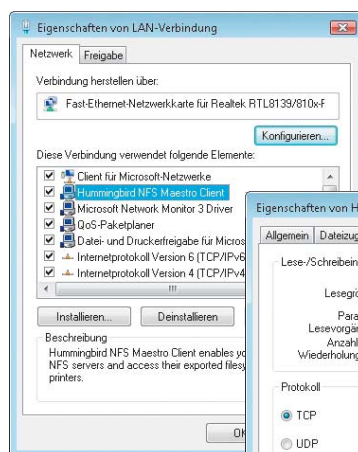
Die aktuelle NFS-Version kombiniert die bisher eigenständigen Teilprotokolle für das Einhängen und die Dateisperren (mount, lock) im NFSv4-Protokoll, sodass sämtliche Daten über einen einzigen Port laufen. NFSv4 antwortet per Vorgabe auf dem TCP-Port 2049. Ältere NFS-Versionen nutzen noch ausschließlich das verbindungslose Transportprotokoll UDP. Das verbindungsorientierte TCP setzt NFSv3 nur auf ausdrücklichen Wunsch ein – erst mit NFSv4 ist es die Vorgabe.

Unter den alten NFS-Versionen exportiert der Server immer Verzeichnisse, die tatsächlich im Dateisystem vorhanden sind. NFSv4 gibt Ressourcen über ein Pseudodateisystem frei, sodass sich beliebige Freigabepfade ohne Umwege definieren lassen und Clients alle Freigaben mit einem einzigen Mount-Befehl einhängen können.

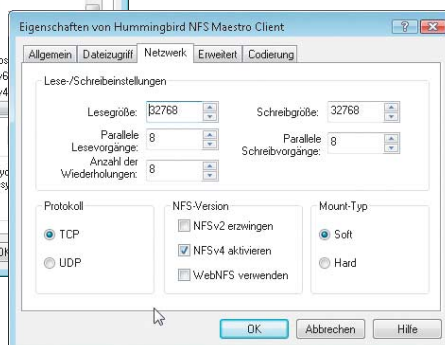
Seit 2003 hat das NFSv4-Protokoll Einzug in aktuelle Betriebssysteme gehalten, die allerdings das Dateisystem und die neuen Sicherheitsfunktionen unterschiedlich gut unterstützen. NFSv4-Clients und -Server stehen mittlerweile auf fast allen Betriebssystemen bereit.

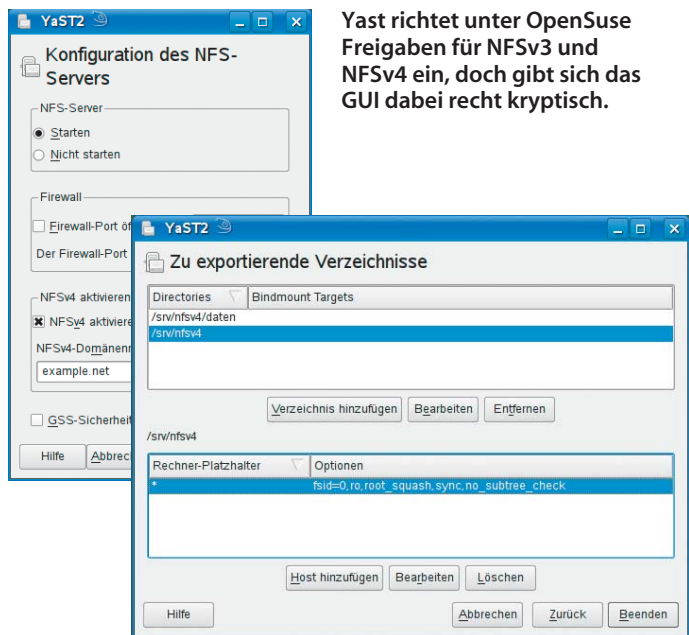
Konkurrenz

Die vielen Sicherheitsprobleme führten dazu, dass NFSv3 nicht als Netzwerkdateisystem für un-



Hummingbirds NFS-Client beherrscht NFSv4, wenn es in den Systemeinstellungen des Clients aktiviert wurde.





YaST richtet unter OpenSuse Freigaben für NFSv3 und NFSv4 ein, doch gibt sich das GUI dabei recht kryptisch.

sichere Netze wie das Internet in Frage kommt. Dort nutzen beispielsweise viele Universitäten das Andrew File System, das sichere Authentifizierung, verschlüsselten Datentransport, Benutzerverwaltung und kryptografische Funktionen an Bord hat.

In lokalen Netzen muss NFS sich mit dem aus der Windows-Welt stammenden Netzwerkdateisystem SMB/CIFS messen, das unter Unixen und Linux in Form des SMB/CIFS-Servers Samba bereitsteht. Es lässt sich vergleichsweise einfach und schnell einrichten, authentifiziert sicher Benutzer und Arbeitsstationen, lässt sich an Verzeichnisdienste wie Active Directory anbinden und besteht nur aus wenigen Programmen. Ab der Version 3.2 verschlüsselt Samba die Verbindungen zwischen Unix-Systemen und verbindet mehrere Einzelserver zu Clustern. Im Vergleich zu NFS bleibt SMB/CIFS jedoch bei der Geschwindigkeit zurück.

Samba steht mittlerweile für viele Betriebssysteme (Mac OS X, Solaris, BSD und Linux) bereit. Microsoft hatte den Samba-Ent-

wicklern Ende 2007 Zugang zu seinen Serverprotokollen gewährt und Anfang 2008 rund 45 000 Seiten Dokumentation zu SMB/CIFS veröffentlicht.

NFSv4 unter Linux

Die erste Referenzimplementierung für Linux stammt von der CITI-Gruppe an der University of Michigan und ist mittlerweile Bestandteil des Linux-Kernels. Seit 2004 versteht der Linux-Kernel (2.6.4) die grundlegenden NFSv4-Operationen sowie die Authentifizierung per Kerberos (krb5/krb5i). Ab Version 2.6.18 verschlüsselt Linux die NFSv4-Anfragen (krb5p) [4]. Der NFSv4-Kernelserver unter Linux liefert Daten sowohl per NFSv3 als auch über die neue Version aus. Welches Protokoll er für Freigaben wählt, bestimmen die Optionen der Einrichtungsdatei (siehe Folgeartikel).

Windows

Für die alte NFS-Version (NFSv3) stellt das Microsoft-Betriebssystem einen eigenen Client bereit, der sich unter XP im Paket „Mi-

crosoft Windows Services for UNIX“ versteckt. Unter Vista (Ultimate/Enterprise) nennt er sich „Subsystem for UNIX-based Applications“ (siehe Soft-Link). Mit NFSv4 versteht er sich unterdessen nicht. Der einzige NFSv4-Client für Windows kommt von Hummingbird, die auch einen NFSv4-Server für Windows anbieten. Der NFS Maestro Client/Solo kostet zwar Lizenzgebühren, kann aber nach einer Anmeldung 60 Tage lang kostenlos getestet werden (siehe Soft-Link).

Die Software klemmt sich an die aktive Netzwerkverbindung, stellt verschiedene Hilfen für die Verbindungseinrichtung bereit und besitzt einen NFS-Browser, der im LAN aktive Server zeigt. Per Vorgabe deaktiviert die Client-Software NFSv4, was sich über die Eigenschaften der aktiven LAN-Verbindung ändern lässt. Der Client authentifiziert sich per Kerberos und LIPKEY.

BSD/Mac OS X

Die University of Michigan hat bereits 2004 einen Kombi-Client für FreeBSD und Mac OS X veröffentlicht, der sich im Repository der BSD-Distribution findet (siehe Soft-Link). Auch für OpenBSD und NetBSD stehen Clients und Server bereit. Für Mac OS X (PPC und Intel) finden sich unter <http://snowwhite.cis.uoguelph.ca/nfsv4> ein NFSv4-Client und dessen Quelltexte. Apple selbst baut bislang nur Clients und Server für NFSv3 in sein Betriebssystem ein.

OpenSolaris

Suns Betriebssystem besitzt dank der Entwicklungsarbeit des Herstellers aktuelle Clients und Server für das Netzwerkdateisystem. Auf OpenSolaris und dem kommerziellen Solaris 10 stehen bereits Programme für die NFS-Version 4.1 bereit, die NFS-Sitzungen, Verzeichnisdelegation und paralleles NFS (pNFS) beherrschen (siehe Soft-Link).

Fazit

Wer im LAN Verzeichnisse ohnehin per NFSv3 an andere Rechner exportiert und dabei auf Kerberos verzichtet, kann gleich aus mehreren Gründen auf NFSv4 umsteigen: NFSv4 schafft das

Wirrwarr der bisherigen Teilprotokolle ab. Sämtliche Funktionen für den Datenzugriff laufen über einen einzigen TCP-Port, was sich leicht mit Firewalls überwachen lässt. Im Unterschied zu NFSv3 ist es nicht zustandslos: Server und Client informieren sich gegenseitig über ihre Aktionen. Hängende Prozesse oder kaputte Daten, wie sie noch unter NFSv3 auftreten konnten, gehören damit der Vergangenheit an. Für einen Wechsel sprechen zusätzlich die flexiblere Einrichtung und die vereinfachte Wartung.

Die neuen NFSv4-Sicherheitsfunktionen verlangen hingegen eine zusätzliche Kerberos-Infrastruktur und teilweise noch weitere Dienste. Den aktuellen Linux-Distributionen fehlen für deren oft komplizierte Einrichtung jedoch geeignete Hilfsmittel oder fertige Pakete, die passende Vorgaben setzen oder bei der Fehlersuche helfen. Eine vollständige NFSv4-Einrichtung samt sicherer Authentifizierung und Verschlüsselung über Kerberos benötigt daher reichlich Handarbeit, die unter Linux schnell in einer Sackgasse enden kann. Mehr Erfolg im Produktionseinsatz versprechen Solaris 10 oder OpenSolaris, die eine ausgereifte NFSv4-Implementierung samt Kerberos-Unterstützung mitbringen.

Ein weiteres Problem stellen die NFSv4-Clients dar, die unter Windows und Mac OS X entweder kostenpflichtig oder nur als Quelltext vorliegen. Wer sicher und unkompliziert Daten zwischen Windows, Mac OS X und Linux tauschen will, dem stellt der NFS-Konkurrent Samba momentan weniger Stolperfallen in den Weg. SMB/CIFS läuft auf unterschiedlichen Betriebssystemen und beherrscht selbst in der Minimaleinrichtung die sichere Benutzeranmeldung. (rek)

Literatur

- [1] IETF-Standard zu NFSv4 (RFC 3530): www.heise.de/netze/rfc/rfc3530.shtml
- [2] NFSv4 Minor 1: <http://tools.ietf.org/html/draft-ietf-nfsv4-minorversion1-28>
- [3] Kerberos 5 (RFC4120): www.heise.de/netze/rfc/rfc4120.shtml
- [4] NFSv4 Linux Features: www.citi.umich.edu/projects/nfsv4/linux/features.html



Den Netzwerkpfad des NFS-Servers bildet der Maestro-Client unter Windows auf einen Laufwerksbuchstaben ab.