

## Distributed File Systems – NFS

Ein DFS ermöglicht dem Benutzer transparenten (!) Zugriff auf das Dateisystem eines anderen Hosts.

### **NFS**

- Erstes verbreitete Netzwerk FS
- Entwickelt Anfang der 80er von Sun
- Stateless Design
- Portabilität durch RPC (Remote Procedure Call)
  - o Protokoll, welches dem Programmierer ermöglicht, Funktionen zu verwenden, die auf einem entfernten Host implementiert sind --> unabhängig vom Transport Protokoll.

### **Versionen**

- NFSv2 (RFC1094)
  - o Ursprüngliches Public Release
  - o Signifikante Verzögerungen bei Schreiboperationen
- NFSv3 (RFC1813)
  - o Verbesserte Performance durch Asynchrones Schreiben (Hierbei wird nicht mehr wie beim synchronen Zugriff ein Arbeitsvorgang komplett abgeschlossen, sondern die gecachten Daten zuerst beim Server zwischengespeichert bis der Client mit einem commit request den Server dazu veranlasst, die gesammelten Veränderungen vorzunehmen. Erst auf die endgültige Bestätigung hin verwirft der Client seine Daten im Cache.
  - o Abwärtskompatibel (durch Zurückfallen auf NFSv2-Protokoll)
- NFSv4 (RFC3530)
  - o In Entwicklung

### **Transportmöglichkeiten**

- RPC über UDP: Bessere Performance im LAN
- RPC über TCP: Bessere Performance im WAN (Implementierung in NFSv3)

### **Stateless Design**

Weder Client noch Server befinden sich jemals in einem Zustand, der abhängig von früheren Informationen ist, jede Transaktion ist eigenständig.

Dies bietet zwei wesentliche Vorteile:

- Bessere Skalierbarkeit des Servers, da keine zusätzlichen Ressourcen pro Client angefordert werden müssen
- Crash Recovery: Falls der Server rebootet, schickt der Client weiterhin den fehlgeschlagenen request bis er wieder eine Antwort vom Server erhält

### **Server-Komponenten**

- portmap: Dienst, bei dem sich alle RPC-fähigen Programme mit ihrem Port registrieren müssen.
- mountd: Überprüft ob der Client berechtigt ist, das angeforderte Verzeichnis zu mounten.
- nfsd: Handled filesystem requests.
- /etc/exports: Enthält „exportierte“ Verzeichnisse und deren export-Optionen (read-only u.ä.).

## Ablauf einer Transaktion

+---+		+-----+	+---+	
	----- mountd port? ----->			(1) Client fragt portmapper des Servers nach Portnummer von mountd
		portmap		(2) Antwort m. entsprechend. Portnummer
	<----- port=923 -----			
		+-----+		
	-- filehandle /mnt/c? -->			(3) Client fragt mountd nach Filehandle des Verzeichnisses /mnt/c
C		mountd	S	(4) Wenn berechtigt, Antwort m. entsprechend. Filehandle
L	<-- filehandle=0xfc894 --	(923)	E	
I		+-----+	R	
E	----- nfsd port? ----->		V	(5) Client fragt portmapper des Servers nach Portnummer von nfsd
N		portmap	E	(6) Antwort m. entsprechend. Portnummer
T	<----- port=2049 -----		R	
		+-----+		
	--- using fh 0xfc894, -->			(7) Client fragt nfsd des Servers unter Verwendung des FH des Verz. nach Filehandle der Datei file.txt
	fh for „file.txt“?	nfsd		(8) Antwort m. entsprechend. Filehandle
		(2049)		
	<-- filehandle=0x8c94 ---			
+---+		+-----+	+---+	

[by bonki]

## Security

Server und Client Hosts müssen für den selben Benutzer die selbe UID verwenden, somit kann/soll NFS nur in vertrauenswürdigen Netzen verwendet werden, da root user auf dem Client Host jede ID annehmen können.

## Beschränkungen

- Antwortzeit nicht garantiert --> ungeeignet für Echtzeit-Anwendungen
- Unterhalb des exportierten FS dürfen keine weiteren mount points liegen
- Maximale Dateigröße:  $2^{32} - 1$  (NFSv2)  
 $2^{64} - 1$  (NFSv3)