Verteilte Dateisysteme

Proseminar
Speicher- und Dateisysteme
Sommersemester 2012
Universität Hamburg

von Hauke Holstein

Inhaltsverzeichnis

Abkürzungsverzeichniss

ACL	Access Control List
AFS	Andrew File System
IETF	Internet Engineering Task Force
NFS	Network File System
PTDB	Protection DataBase
RPC	Remote Procedure Call
SMB	Server Message Block
UDP	User Datagram Protocol
VLDB	Volumen Database
WAN	Wie Area Network

1 Einleitung

Ein verteiltes Dateisystem ermöglicht den Zugriff auf Dateien über ein Netzwerk. Dabei hat der Client keinen direkten Zugriff auf die zugrunde liegende Blockstruktur der Datenträger. Genauso ist es vollkommen unerheblich, auf was für einem Dateisystem die Daten sich auf dem Server befinden. So ist es einem Programm auf dem Client möglich Dateien so zu behandeln als liegen sie lokal auf dem Rechner ohne sich um die Übertragung und besondere Bedingungen anderer Dateisystem zu kümmern.
Im Folgenden werde ich näher auf drei Implementation verteilter Dateisystem eingehen. Zuerst auf das unter Unix weitverbreitete NFS. Anschließend auf AFS, das in weltumspannenden Netzen eingesetzt wird. Abschließend werde ich das unter Windows übliche SMB beschreiben.

2 Grundlagen

2.1 Zustandslosigkeit

Ein zustandsloses Protokoll behandelt jede Anfrage einzeln ohne Sitzungsinformationen zu speichern. So müssen bei jeder Anfrage alle für die Ausführung nötigen Informationen übertragen werden. Dies hat den Vorteil, das der Server weniger stark belastet wird und im Falle einer unterbrochenen Verbindung die übrig bleibenden Sitzungen nicht aufgeräumt werden müssen. Ein Nachteil dieses Verfahrens ist, das viele Informationen möglicherweise mehrfach übertragen werden müssen.

2.2 Systemaufrufe

Systemaufrufe sind Methoden um vom Betriebssystem bereitgestellte Funktionen ausführen zu können. Dazu zählt der komplette Befehlssatz der CPU als auch Befehle um auf Datenträger zuzugreifen. Letztere sind im Bezug auf Dateisysteme interessant. Systemaufrufe sind nötig, da Anwendungen in der Regel in einem unterprivilegierten Modus laufen und somit keinen direkten Zugriff auf entsprechende Funktionen haben.

3 Network File System

3.1 Entwicklung

Die erste Version von NFS wurde in den 1980er Jahren von Sun entwickelt. Damals wurde es nur intern zu Testzwecken eingesetzt.
Nach einigen grundlegenden Änderungen veröffentlichte Sun NFSv2 im Jahr 1989. In dieser Version verlief die Datenübertragung Zustandslos über UDP.
1995 wurde NFS auf Version 3 aktualisiert. Damit wurde unter anderem die 2GB Dateigrößenbeschränkung der Vorgänger Version entfernt.
Nach der Übergabe, der Entwicklung an die Internet Engineering Task Force erschien im Jahr 2000 NFSv4. Diese Version wurde von AFS und SMB beeinflusst. Während der Entwicklung wurde mehr Wert auf die Sicherheit des Protokolls gelegt. Außerdem unterstützt NFS mit dieser Version die Übertragung über TCP. In diesem Zuge wurde auf zustandsbehaftete Dateizugriffe umgestellt. Mit diesen Änderungen ist NFS besser im WAN und hinter Firewalls nutzbar.
Heute wird NFS hauptsächlich im Unix Umfeld eingesetzt.

3.2 Eigenschaften

NFS ermöglicht es auf entfernte Dateisysteme zuzugreifen, ohne die Dateien auf den eigenen Rechner zu übertragen. Dabei werden die Systemaufrufe an den Server weitergeleitet. Dort werden sie jedoch nicht direkt umgesetzt. Um die Belastung des Servers gering zu halten, wird zum Beispiel der open Aufruf in lookup umgewandelt. Dieser liefert Informationen zum Speicherort der Datei. Bei den folgenden read und write Aufrufen werden diese Informationen angehängt. Dadurch wird die Datei auf dem Server nicht geöffnet und der Server muss keinen Zustand speichern. Dies sorgt für einen sehr Ressourcen sparenden Betrieb.
Auf Client Seite werden nur für kurze Zeit Verzeichnisinformationen und Dateiattribute gespeichert.
Damit eignet sich NFS gut für das Ausführen von Anwendungen und Bearbeiten von Dokumenten im lokalen Netzwerk auch auf leistungsschwachen Servern. Erst mit Einführung von NFSv4 und der Beschränkung auf einen festen Port funktioniert dies auch hinter Firewalls zuverlässig. Allerdings kann das gemeinsame Nutzen von Dateien zu Problemen führen.

3.3 Sicherheit

Beim Zugriff auf eine NFSv3 Server authentifiziert sich nicht der User sonder der Rechner, auf dem der Client läuft. Dazu wird auf dem Server eine Datei verwaltet in der alle IP-Adressen mit Zugrifferlaubnis eingetragen sind. Dies ist auf den Ursprung von NFS zurückzuführen. In den 80er Jahren wurden lokale Netze noch als Sicher angesehen und eine Authentifizierung über die IP-Adresse war völlig ausreichend. Sun hatte vorgesehen die Sicherheit der RPC Schicht zu überlassen. Dabei sollte RPC durch Secure-RPC ersetzt werden. Dies scheiterte allerdings an der geringen Verbreitung von Secure-RPC.
Ist die Verbindung zum Server erst mal hergestellt erfolgt die Zugriffsbeschränkung auf Dateien und Verzeichnisse durch Unix-Dateiattribute. Dies führt zu Problemen in Netzwerken ohne zentrale Benutzerverwaltung, weil die gleiche User ID auf verschiedenen Rechnern unterschiedlichen Benutzern zugeordnet sein kann. Um dieses Problem etwas zu mildern, ist es möglich root Zugriffe zu unterbinden. Damit muss ein Angreifer so lange User IDs durchprobieren, bis er einen Benutzer findet, der Zugriff auf die gewünschte Datei hat.
NFSv4 löst dieses Problem mit der Integration von Kerberos. Kerberos ermöglicht die Authentifizierung durch Benutzer und die Verschlüsselung des Datenverkehrs zwischen Client und Server. Außerdem schafft es ein Vertrauensverhältnis zwischen Client und Server, da sich nicht nur der Client am Server ausweißt, sondern auch umgekehrt. Dadurch kann der Client sicher sein das er wirklich mit dem gewünschten Server verbunden ist und nicht umgeleitet wurde. Durch die zentrale Benutzerverwaltung von Kerberos wird außerdem sichergestellt das die Benutzerrechte nicht umgangen werden können.
Der Nachteil dieser Absicherung ist allerdings ein erhöhter Installationsaufwand.

4 Andrew File System

4.1 Entwicklung

Seine Anfänge nahm AFS an der Carnegie-Mellon University als universitäres Projekt. Daher stammt auch der Name Andrew, mit dem der Gründer der Universität gewürdigt werden sollte. Später übernahm die Firma Transarc die Entwicklung und vermarktete es unter dem Namen Transarc AFS. Nach der Übernahme von Transarc durch IBM führten diese die Entwicklung fort und veröffentlichten es im Jahr 2000 mit dem Namen Open AFS unter einer Open Source Lizenz. Neben Open AFS gibt es noch viele weitere Implementationen von AFS.
Als Server Betriebssystem wird bei Open AFS Linux, AIX sowie Solaris unterstützt. Der Client läuft unter Linux, Windows, OS X, AIX und Solaris.

4.2 Architektur

Anders als NFS wo ein Client auch gleichzeitig Server sein kann wird bei AFS strickt zwischen Client und Server getrennt. Pro AFS Zelle muss es mindestens einen Datenbank- und Dateiserver geben.
AFS ist darauf ausgelegt Dateien weltweit und Netz übergreifend zur Verfügung zu stellen. Es skaliert gut mit einer sehr großen Anzahl von Benutzern.
Greift ein Client auf eine Datei zu fragt dieser zuerst beim Datenbankserver nach auf welchem Dateiserver sich die Datei befindet. Anschließend wird die Datei vom Dateiserver auf den Client übertragen und Lokal gespeichert. Wird die lokale Kopie verändert, meldet der Client beim Schließen der Datei die Veränderung an den Dateiserver. Dieser meldet allen erreichbaren Clients, die ebenfalls eine Kopie dieser Datei haben, dass eine Änderung vorliegt. Durch die lokale Zwischenspeicherung der Daten sind auch im Betrieb übers WAN niedrige Zugriffszeiten garantiert.
Die Aufteilung in Datenbank- und Dateiserver ermöglicht es im laufenden Betrieb einzelne Dateiserver auszutauschen oder weiter Dateiserver hinzuzufügen. Auch bei vielen Dateiservern entsteht für den Client der Eindruck eines einzigen großen Dateisystems.
AFS sorgt nicht nur für die Verteilung der Daten, es pflegt auch eine eigene Benutzerverwaltung, die auf Kerberos basiert. Außerdem umfasst es Werkzeuge zur Datensicherung und Synchronisation der Uhrzeit.

4.2.1 Datenbankserver

Die Datenbankserver sind untereinander vernetzt und verwalten die Protection Database (PTDB) und die Volumen Database (VLDB). Neben diesen beiden Datenbanken sind auch noch weitere möglich. Um in einer Datenbank schreiben zu können, muss mehr als die Hälfte der Datenbankserver erreichbar sein. Für lesenden Zugriff ist nur ein Server notwendig. So können einzelne Server ausgetauscht werden ohne das der Schreibzugriff auf die Datenbanken eingebüßt wird.
In der Protection Database werden Benutzer und Gruppen abgelegt. Die Volumen Database speichert die IP-Adressen der Dateiserver und listet alle Volumens auf.

4.2.2 Dateiserver

Ein Dateiserver ist in verschiedene Volumes aufgeteilt. Diese werden vom Administrator angelegt und sind in ihrer Größe veränderbar. Die Volumes können in verschiedenen Arten auftreten.
RW-Instanzen sind Volumes auf die sowohl lesend als auch schreibend zugegriffen werden kann. Sie können im laufenden Betrieb zwischen Dateiservern verschoben werden.
RO-Instanzen sind Kopien von RW-Instanzen. Es ist möglich von einer RW-Instanz beliebig viele RO-Instanzen zu erstellen. Dadurch kann die Leistung deutlich erhöht werden. Clients verbinden sich immer mit dem ihnen am nächsten gelegenen funktionierenden Dateiserver. Dadurch ergeben sich kürze Übertragungszeiten, und falls einzelne Dateiserver ausfallen, bleiben die Daten erreichbar, solang eine RO-Instanz verfügbar ist.
Eine Backup-Instanz befindet sich immer auf demselben Datenträger wie die zugehörige RW-Instanz. Anders als der Name vermuten lässt, ist sie daher kein physikalisches Backup. Diese Volumes werden vom AFS Backup System dazu benutzt Images von RW-Instanzen zu erstellen.
Die letzte Volume Art sind die temporären Clones. In der Regel bekommt niemand etwas von ihrem Auftauchen mit. Sie sorgen dafür das beim Verschieben von Volumes der Schreibzugriff erhalten bleibt.

4.3 Sicherheit

Jede AFS Zelle hat einen 56 bit breiten Schlüssel der allen AFS Servern bekannt ist. Dieser Schlüssel ist ebenfalls dem Kerberos Server bekannt und gewährleistet somit das Benutzer authentifiziert werden können.
Zusätzlich werden Datenübertragungen mit einem 56 bit breiten Sitzungsschlüssel signiert und können bei Bedarf verschlüsselt werden.
Unter aktuellen Gesichtspunkten ist diese Absicherung wegen des nur 56 bit breiten Schlüssels nicht mehr besonders stark. Außerdem kann die gesamte Zelle kompromittiert werden, wenn ein einzelner Server von Unbekannte übernommen wird.

4.4 Zugriffsrechte

Bei AFS werden die Zugriffsrechte über sogenannte ACLs (Access Control List) definiert. Mit diesen Listen können die Rechte für einzelne Verzeichnisse festgelegt werden. Diese Rechte gelten dann automatisch für darin enthaltene Dateien und neu erstellte Verzeichnisse. Es können für jeden Benutzer unterschiedliche Rechte gesetzt werden. Außerdem lassen sich Benutzer in Gruppen zusammenfassen.
Folgende Berechtigungen sind unter AFS möglich:

- r (read): Dateien innerhalb des Verzeichnisses können gelesen und Ausgeführt werden. Anders als unter Unix üblich besitzt AFS keine extra Berechtigung für das Ausführung von Dateien.
- w (write): Dateien dürfen verändert werden.
- l (lookup): Das Verzeichnis darf gelesen werden.
- i (insert): In dem Verzeichnis dürfen neue Dateien erstellt werden.
- d (delete): Es dürfen Dateien gelöscht werden.
- k (lock): Dateien können gesperrt werden.
- a (administer): Die ACL des Verzeichnisses darf verändert werden.

5 Server Message Block

5.1 Entwicklung

Ursprünglich wurde SMB 1983 bei IBM entwickelt. Nach und nach wurde es von verschiedenen Firmen erweitert. Die meisten Erweiterungen steuerte Microsoft bei. Sie legten aber erst im Jahr 2007 vollständige Dokumentationen offen, nachdem die EU Druck ausgeübt hatte. Mit den letzten drei Windows Versionen veröffentlichte Microsoft auch jeweils neue Version von SMB. Zuletzt wurde mit Windows 8 SMB 3.0 eingeführt. Neben der Implementation von Microsoft wurde 1992 Samba veröffentlicht. Es ermöglicht den Zugriff von Unix Systemen auf Windows Freigaben und umgekehrt.

5.2 Eigenschaften

SMB stellt neben Dateifreigaben auch Druck- und andere Serverdienste zur Verfügung. Dabei erfolgt die Datenübertragen über einen festen TCP-Port. Ähnlich wie NFS kann ein Client auch gleichzeitig Server sein, deshalb eignet es sich gut für kleinere Netzwerke. Aufgrund der Verschiedenen sich im Einsatz befinden SMB Versionen müssen sich Client und Server vor Beginn der Datenübertragung auf eine gemeinsame Befehlsbasis einigen, diese Basis wird auch als Dialekt bezeichnet.

5.3 Sicherheit

Zur Authentifizierung nutzt SMB eine eigene Benutzerliste. Außerdem lässt es sich an Kerberos koppeln. Unter Windows nutzt SMB die Windows Benutzerkonten oder Benutzer aus einer Active Directory.
Im Laufe der Jahre sind immer wieder Sicherheitslücken in Microsofts SMB Implementierung aufgetaucht.

Quellen

Allgemein

http://en.wikipedia.org/wiki/Distributed_file_system
http://de.wikipedia.org/wiki/Verteiltes_Dateisystem
http://www.informatik.uni-leipzig.de/~irmscher/lehre/skripte/VerteilteSystemeScriptum.pdf
http://en.wikipedia.org/wiki/Stateless_protocol
http://de.wikipedia.org/wiki/Zustandslos

NFS

http://en.wikipedia.org/wiki/Network_File_System
http://de.wikipedia.org/wiki/Network_File_System
http://www.heise.de/netze/artikel/Das-Netzwerk-Dateisystem-NFSv4-221577.html

AFS

http://www.feyrer.de/SA/vortraege/ss2008-troppmann-paper.pdf
http://de.wikipedia.org/wiki/Andrew_File_System#OpenAFS
http://www.urz.uni-heidelberg.de/datenhaltung/afs/

SMB

http://de.wikipedia.org/wiki/Server_Message_Block
http://en.wikipedia.org/wiki/Server_Message_Block
http://de.wikipedia.org/wiki/Samba_(Software)