

# AFS – ein verteiltes Dateisystem

Thomas Müller (thomas.mueller@hrz.tu-chemnitz.de),  
Tino Schwarze (tino.schwarze@informatik.tu-chemnitz.de)

10. März 2002

## Zusammenfassung

AFS ist ein leistungsfähiges verteiltes Dateisystem mit langer Tradition. Mit der Veröffentlichung der AFS-Sourcecodes als OpenAFS ist es seit Oktober 2000 für jedermann verfügbar.

Dieser Workshop zeigt, wie man AFS-Server und die zugehörigen Clients installiert, um sein eigenes Netzwerk mit AFS aufzuwerten.

# Geschichte von AFS

- **ab 1984:** an Carnegie Mellon University als „Andrew File System“ entwickelt
- **1989:** Gründung von Transarc zur kommerziellen Verwertung, Umbenennung in AFS
- **1998** IBM übernimmt Transarc
- **15.08.2000:** IBM kündigt an, AFS als OpenSource zu veröffentlichen
- **31.10.2000:** OpenAFS 1.0 wird veröffentlicht.
- **30. Januar 2002:** OpenAFS 1.2.3 wird veröffentlicht.

# Eigenschaften von AFS (1)

- verteiltes Filesystem, vergleichbar mit NFS
- globaler Namensraum (/afs)
- mehrere Server, transparenter Zugriff über Pfadname
- Volumemanagement, Migration von Volumes zwischen Servern
- mehrere redundante read-only Kopien (Clones) von Volumes möglich
- effizientes Caching beim Client

## Eigenschaften von AFS (2)

- hohe Sicherheit (Kerberos-Authentifizierung, Verschlüsselung)
- *Access-Control-Lists* (ACLs) auf Verzeichnisebene
- Quota auf Volumebasis
- Backup im laufenden Betrieb
- direkter Zugriff auf das Backup

# Terminologie (1)

**Zelle** Unabhängige Einrichtung, die ein AFS mit eigenem Namensraum betreibt. Zellen werden über einen Namen identifiziert. Empfehlung: Internet-Domainname.

Computer gehören üblicherweise einer „Heimatzelle“ an, Nutzer können aber Accounts in mehreren Zellen haben (und diese auch parallel nutzen). Eine Zelle kann auch geographisch verteilt sein. Der Zugriff auf fremde Zellen erfolgt mit Hilfe spezieller DNS-Einträge oder durch eine manuell gepflegte Datenbasis.

**Volume** Ein Container für eine Menge von Dateien, in der Größe durch eine *Quota* beschränkt. Volumes werden im globalen Namensraum mit Hilfe von Mountpoints montiert – das gleiche Volume auch mehrfach. Volumes können repliziert und transparent auf andere Fileserver migriert werden.

## Terminologie (2)

**Partition** Speicherplatz auf dem Server, nimmt Volumes auf.

**Mount Point** Vergleichbar mit einem symbolischen Link, ordnet einem Verzeichnis ein Volume zu.

**Replikation** Anlegen von read-only Kopien eines Volumes zwecks höherer Verfügbarkeit. Sinnvoll für hoch frequentierte Volumes, deren Inhalt sich selten ändert, z. B. Bibliotheken und Applikationen.

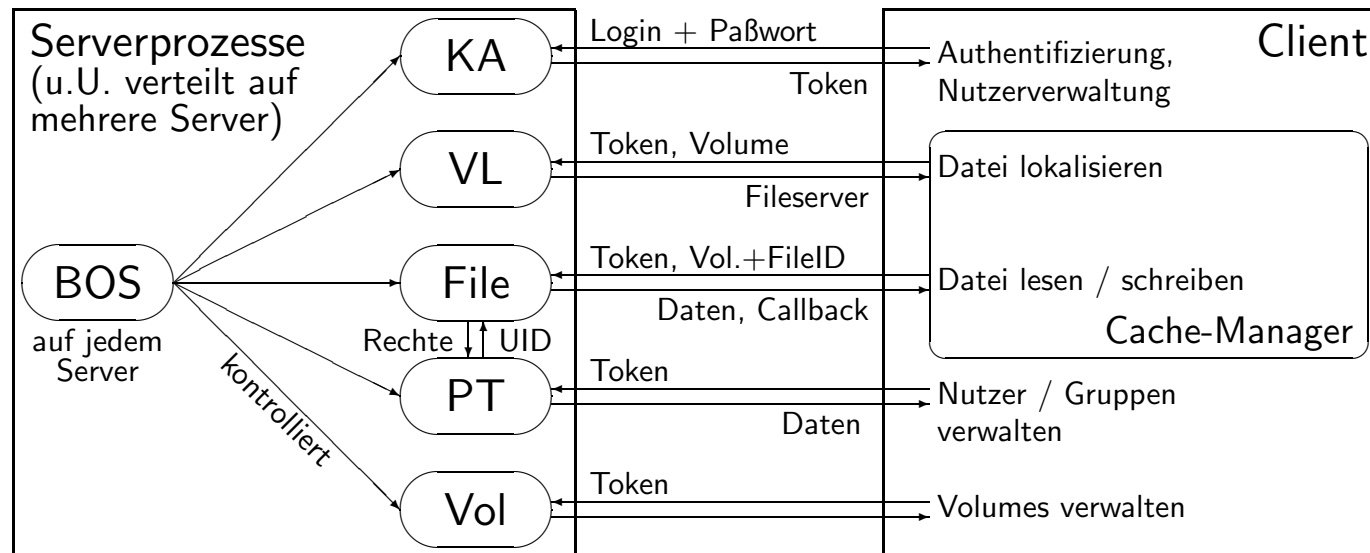
## Terminologie (3)

**Caching, Callback** Der AFS-Client cachet Dateien, die im AFS liegen, auf der lokalen Festplatte. Wird eine gecachte Datei auf dem Server verändert, informiert dieser den Client darüber mittels eines *Callback*.

**AFS-Nutzer** Nutzerinformationen werden bei AFS in der KADB (Authentifizierung) und PTDB (Abbildung Nutzernamen auf UID, Gruppen) verwaltet. Zusätzlich sind noch Informationen aus `/etc/passwd` notwendig (z.B. Homeverzeichnis, voller Name, Login-Shell).

## Wie sieht ein AFS-Server aus?

Es gibt nicht *den* AFS-Server. Die verschiedenen Dienste werden auf einzelne Prozesse verteilt, die wiederum auf verschiedenen Maschinen laufen können. Alle Serverdienste können von jedem AFS-Client aus administriert werden.



## Die wichtigsten Serverprozesse

**BOS Server** *Basic OverSeer Server*. Überwacht die einzelnen Serverprozesse.

**File Server** Liefert Dateien aus und nimmt Änderungen entgegen.

**Volume Server** Verwaltung der Volumes. Erstellen, Migrieren, Löschen.

**Authentication Server** Verwaltet die Kerberos-Authentifizierungsdatenbank (Logins, Passwörter, Schlüssel), verifiziert die Identität der Nutzer.

**Protection Server** Zuordnung von Login zu UID, Gruppenverwaltung

**Volume Location Server** Verwaltet die *Volume Location Database* (VLDB).

## Weitere Serverprozesse

**Update Server** Verteilung von Updates wichtiger Konfigurationen oder der Server Software über mehrere Server.

**Backup Server** Automatisches Backupsystem.

**Salvager** Kein Server als solcher. Wird vom BOS Server aufgerufen, wenn der File- oder Volume-Location-Server ausfällt oder vermutlich inkonsistente Platten vorliegen.

**NTPD** Network Time Protocol Daemon für Zeitsynchronisierung.

**Cache Manager** Läuft auf dem Client, bietet das Interface zu den AFS-Servern.

# Vorgehensweise zur Serverinstallation

- Linux installieren, mind. 1 Partition freihalten als AFS-Partition
- Software beschaffen
- OpenAFS–Software installieren
- AFS–Serverdienste konfigurieren und starten (Teil 1)
- Client konfigurieren und starten
- AFS–Serverdienste konfigurieren (Teil 2)

# OpenAFS–Bezugsquellen

1. von der OpenAFS-Homepage

<http://www.openafs.org/release/latest.html>

2. vom Mirror an der TU Chemnitz

<ftp://ftp.tu-chemnitz.de/pub/openafs/>

3. zusätzlich für die TU angepaßte RPMs für RedHat und SuSE

<http://www.tu-chemnitz.de/urz/afs/openafs/>

# OpenAFS–Pakete installieren

- Linux haben wir schon einmal vorbereitet: RedHat 7.2
- OpenAFS–RPMs einspielen (Basispaket und Serverpaket):

```
> rpm -i openafs-1.2.3-rh7.2.2.i386.rpm \  
openafs-server-1.2.3-rh7.2.2.i386.rpm \  
openafs-client-1.2.3-rh7.2.2.i386.rpm \  
openafs-kernel-1.2.3-rh7.2.2.i386.rpm \  
openafs-kpasswd-1.2.3-rh7.2.2.i386.rpm
```

## AFS–Serverdienste Teil 1.0

- Filesystem für `/vicepa` erzeugen und mounten
- BOSServer starten:  
> `/usr/afs/bin/bosserver -noauth`
- Name für AFS–Zelle festlegen:  
> `bos setcellname <fileserver> clug.de -noauth`
- Inhalt von `/usr/afs/etc/CellServDB` des Servers in `/usr/vice/etc/CellServDB` auf dem Client übernehmen.
- Auf dem Client Zellename in `/usr/vice/etc/ThisCell` eintragen.

# AFS-Serverdienste Teil 1.1

- Datenbankserver starten:

- > bos create localhost kaserver simple /usr/afs/bin/kaserver \  
-noauth
- > bos create localhost ptserver simple /usr/afs/bin/ptserver \  
-noauth
- > bos create localhost vlserver simple /usr/afs/bin/vlserver \  
-noauth

## AFS-Serverdienste Teil 1.2

- Notwendige Einträge in Kerberos-Datenbank erzeugen:

```
> kas -cell clug.de -noauth
```

```
ka> create afs (Kerberos-Dienst „afs“ anlegen)
```

```
ka> create admin (Nutzer admin anlegen)
```

```
ka> setfields admin -flags admin (Nutzer admin ist root-Äquivalent)
```

```
ka> quit
```

- Nutzer „admin“ Recht auf privilegierte AFS-Kommandos geben:

```
> bos adduser localhost admin -noauth
```

## AFS-Serverdienste Teil 1.3

- Server-Schlüssel hinterlegen
  - > `bos addkey localhost -kvno 0 -noauth`
  - > `bos listkeys localhost -noauth`
- PTS-Datenbankeintrag für Nutzer „admin“
  - > `pts createuser -name admin -id 1 -noauth`
  - > `pts adduser admin system:administrators -noauth`
- BOSServer neu starten:
  - > `bos restart localhost -all -noauth`

## AFS-Serverdienste Teil 1.4

- Fileserver starten:  
> bos create localhost fs fs /usr/afs/bin/fileserver \  
/usr/afs/bin/volserver /usr/afs/bin/salvager -noauth
- Volume root.afs anlegen:  
> vos create localhost a root.afs -noauth
- BOSServer stoppen  
> bos shutdown localhost -wait -noauth  
> killall bosserv
- ab jetzt wird authentifiziert gearbeitet

## AFS-Client konfigurieren

- Zur weiteren Installation wird ein AFS-Client benötigt, da bereits im /afs-Verzeichnisbaum operiert werden muss.
- Software auf Client installieren (falls noch nicht geschehen)
  - > rpm -i openafs-1.2.3-rh7.2.2.i386.rpm
  - > rpm -i openafs-client-1.2.3-rh7.2.2.i386.rpm\  
openafs-kernel-1.2.3-rh7.2.2.i386.rpm
- /etc/sysconfig/afs anpassen (AFS\_CLIENT=on, AFS\_SERVER=on, Cache)
- AFS-Client und Server starten:
  - > /etc/init.d/afs start

# Rechte im AFS

- AFS unterscheidet einige Rechte mehr als ein Standard–Unix:

l	lookup	Verzeichnisinhalt einsehen
r	read	Dateiinhalt lesen
i	insert	Dateien anlegen
w	write	Dateien beschreiben
d	delete	Dateien löschen
k	lock	Dateien sperren
a	admin	Rechte ändern

- Für jedes Verzeichnis kann eine *Access Control List* (ACL) gesetzt werden, die einzelnen Nutzern und/oder Gruppen Rechte zuweist oder aberkennt.

## AFS-Serverdienste Teil 2

- Als AFS-Superuser anmelden:  
> klog admin
- AFS-Volume root.cell erzeugen:  
> vos create <fileserver> a root.cell  
> fs setacl /afs system:anyuser l
- Mountpoints für Volume root.cell anlegen:  
> fs mkmount /afs/clug.de root.cell  
> fs setacl /afs/clug.de system:anyuser rl

# Volume-Verwaltung

- Volumes anlegen:  
> `vos create <fileserver> <partition> <volume>`
- Volume montieren (evtl. Read/Write-Volume erzwingen)  
> `fs mkmount <pfad> <volume> [-rw]`
- Volume zwischen Fileservern oder Partitionen verschieben  
> `vos move <volume> <fromserver> <frompartition> \  
    <toerver> <topartition>`

# Volume-Replikation

- Bei mehreren Servern können häufig gelesene Volumes repliziert werden, um Flaschenhälse zu vermeiden. Gute Kandidaten z. B.: `root.afs`, `root.cell`  
Existieren von einem Volume read-only Kopien, werden diese bevorzugt verwendet.
- Anlegen einer RO-Kopie: (`<volume>.readonly`)  
> `vos addsite <fileservers> <partition> <volume>`
- Synchronisieren aller RO-Kopien mit dem RW-Volume:  
> `vos release <volume>`

# Nutzer anlegen

- Nutzer anlegen:
  - > kas create <login> -admin <administrationsaccount>
  - > pts createuser <login> [<uid>]
- Es empfiehlt sich, die AFS-UIDs identisch zu den System-UIDs zu wählen.
- Es können auch IP-Nummern als Nutzer angelegt werden. Diese müssen dann aber zu einer Gruppe gehören, um in ACLs verwendbar zu sein.
- Da Anlegen von Nutzern recht komplex ist, empfiehlt sich die Einführung von Automatismen (z.B. `uss`)

# AFS–Authentifizierung auf dem Client

- Die Nutzerdatenbank im AFS soll natürlich zur Authentifizierung verwendet werden. AFS verwaltet leider keine Nutzer–Metadaten wie Homeverzeichnis, Name etc. Diese müssen weiterhin per `/etc/passwd` o.ä. zur Verfügung gestellt werden.
- Es gibt für einige Programme AFS–Modifikationen, i. A. bietet sich aber die Verwendung von PAM an. Entsprechende Module sind Bestandteil der OpenAFS–Releases. Zusatz-Einträge für `/etc/pam.d/*`:

```
auth    sufficient /lib/security/pam_afs.so.1 ignore_root try_first_pass
account sufficient /lib/security/pam_afs.so.1 ignore_root try_first_pass
session optional  /lib/security/pam_afs.so.1 ignore_root
```

# Gruppen anlegen

- Die Protection Database (PTDB) verwaltet auch die AFS-Gruppen. Hier spielt die KADB keine Rolle.
- Jeder AFS-Nutzer darf auch eigene Gruppen anlegen, diese müssen dann aber `<login>:<gruppenname>` heißen.
- Neue AFS-Gruppe anlegen, Nutzer eintragen, Mitglieder anzeigen:
  - > `pts creategroup <groupname>`
  - > `pts adduser <login> <groupname>`
  - > `pts membership (<groupname>|<login>)`

## Rechte und Quota im AFS ändern

- Einem Nutzer/einer Gruppe Rechte einräumen oder explizit aberkennen:  
> fs setacl <verzeichnis> (<login>|<gruppe>) <rechte>  
> fs setacl <verzeichnis> (<login>|<gruppe>) <rechte> \  
-negative
- Für <rechte> kann man auch die Abkürzungen none, read, write oder all verwenden.
- Quotas gelten pro Volume.
- Die Quota für ein Volume einsehen oder ändern:

```
> fs listquota <verzeichnis>  
> fs setquota <verzeichnis> <quota_in_kb>
```

# Abschluß

Der Workshop sollte einen Überblick zu AFS geben und aufzeigen, dass der Aufwand zum Einrichten einer AFS-Zelle beherrschbar ist. Die Vorteile des AFS-Einsatzes werden vor allem bei der täglichen Arbeit - ob aus Nutzer- oder aus Admin-Sicht - offensichtlich.

Einige Themen wurden nicht oder nur am Rande angesprochen (z.B.: Organisation der Datensicherung, Automatisierung von Vorgängen: Nutzerverwaltung, etc.). Solche Mechanismen gibt es, ihre Darstellung würde den Rahmen dieses Workshops sprengen.

## Nachtrag: Vorführung

Der Vollständigkeit halber hier ein grobes Protokoll der Live-Demonstration.  
(Anmerkung: gaheris und galahad sind Rechnernamen, das Einrichten der Zelle wurde auf dem Rechner gaheris begonnen)

```
# Anlegen diverser Volumes:
```

```
vos create gaheris b home  
fs mkmount /afs/clug.de/home home  
fs copyacl /afs/clug.de \  
    /afs/clug.de/home
```

```
vos create gaheris b software  
fs mkmount /afs/clug.de/software software  
fs copyacl /afs/clug.de \  
    /afs/clug.de/software
```

```
    /afs/clug.de/software
# Migrieren eines Volumes:
vos move software gaheris b gaheris a

### galahad einrichten inkl. client ###
# RPMs einspielen

# Hier sollte besser der Updateserver verwendet werden!
scp -r /usr/afs/etc galahad:/usr/afs/etc
scp /usr/vice/etc/{ThisCell,CellServDB} galahad:/usr/vice/etc

# client+server aktivieren, Cache
vi /etc/sysconfig/afs

# client + server starten
```

```
/etc/init.d/afs/start
```

```
# Fileserverprozesse erzeugen
```

```
bos create galahad fs fs /usr/afs/bin/fileserver \  
    /usr/afs/bin/volserver /usr/afs/bin/salvager
```

```
# root.{cell,afs} sollte auf jedem Fileserver verfuegbar sein
```

```
fs mkmount /afs/.clug.de root.cell -rw
```

```
vos addsite gaheris b root.afs
```

```
vos addsite galahad a root.afs
```

```
vos examine root.afs
```

```
vos release root.afs
```

```
vos examine root.afs
```

```
vos addsite gaheris b root.cell
```

```
vos addsite galahad a root.cell
vos examine root.cell
vos release root.cell
vos examine root.cell
```

```
# Nutzer anlegen
```

```
vos create galahad a home.tisc
fs mkmount /afs/clug.de/home/tisc home.tisc
kas create tisc -admin_user admin
pts createuser tisc -id 16634
fs setacl /afs/clug.de/home/tisc tisc all
```

```
#useradd -c "Tino Schwarze" -d /afs/clug.de/home/tisc \
# -m -u 16634 tisc
```

```
# evtl.  
#vos create galahad a home.thm  
#fs mkmount /afs/clug.de/home/thm home.thm  
#kas create thm -admin_user admin  
#pts createuser thm -id 4753  
#fs setacl /afs/clug.de/home/thm thm all  
#useradd -c "Thomas Mueller" -d /afs/clug.de/home/thm \  
# -m -u 4753 thm
```

```
# jetzt PAM einrichten  
auth sufficient /lib/security/pam_afs.so.1 \  
ignore_root try_first_pass  
account sufficient /lib/security/pam_afs.so.1 \  
ignore_root try_first_pass
```

```
session optional /lib/security/pam_afs.so.1 \
ignore_root

# login als tisc mittels gdm
cd /afs/clug.de/home/tisc

cp /usr/src/redhat/RPMS/i386/openafs-* . # Quota reicht nicht
fs listquota /afs/clug.de/home/tisc

# in zweiter Shell
klog admin -setpag
fs setquota /afs/clug.de/home/tisc 500000

# als tisc
```

```
cp /usr/src/redhat/RPMS/i386/* .  
# parallel als admin  
fs move home.tisc gaheris a galahad a  
  
# Backup-Volume anlegen und im Homeverzeichnis montieren  
vos backup home.tisc  
fs mkmount BACKUP home.tisc.backup
```