AFS – ein verteiltes Dateisystem

Chris Hübsch (chris.huebsch@informatik.tu-chemnitz.de),
Tino Schwarze (tino.schwarze@informatik.tu-chemnitz.de)

9. November 2001

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 Vortrag soll zeigen, wie man einen 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, Umbennenung 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.
- **Ende Oktober 2001:** OpenAFS 1.2.2 wird veröffentlicht.

Eigenschaften von AFS (1)

- Netzwerkfilesystem ähnlich 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 Namensbereich betreibt.

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. Eine globale, manuell gepflegte Datenbank ermöglicht den Zugriff auf externe Zellen.

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.

Partition Speicherplatz auf dem Server, nimmt Volumes auf.

Terminologie (2)

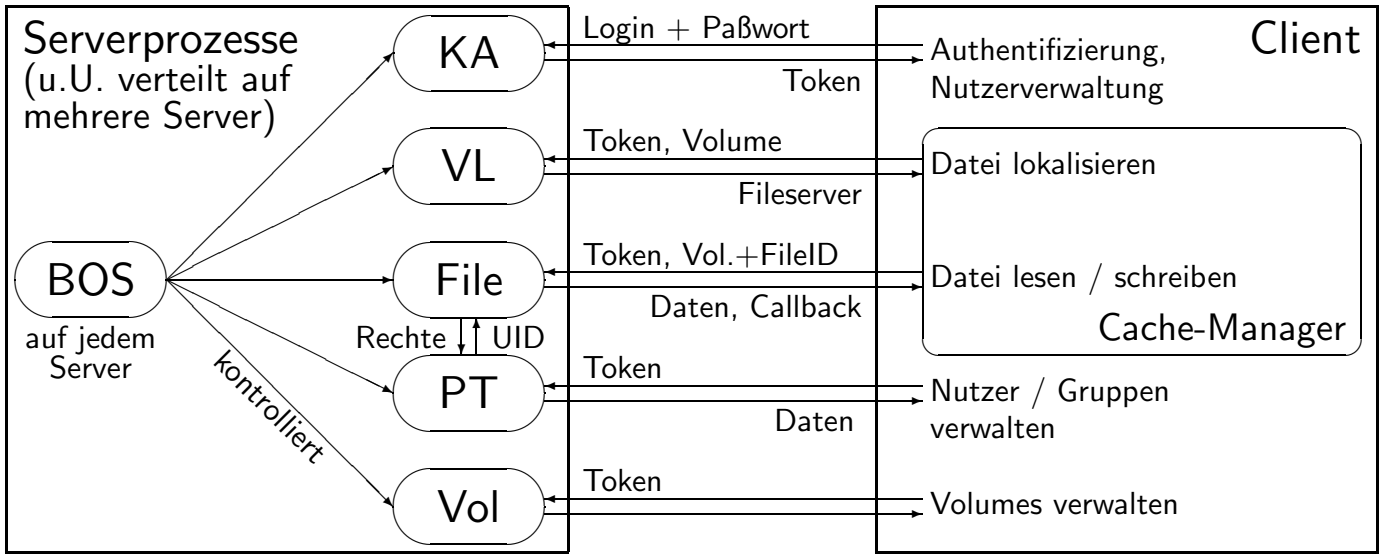
Mount Point Vergleichbar mit einem symbolischen Link, ordnet einem Verzeichnis ein Volume zu.

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.

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

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.

Authentication Server Verwaltet die Kerberos–Authentifizierungsdatenbank (Logins, Passwörter, Schlüssel), verifiziert die Idendität der Nutzer.

Protection Server Verwaltet die Zugriffsrechte im AFS (Gruppen, ACLs).

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

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 Volme–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.

Vorgehen zur Serverinstallation

- Linux installieren, mind. 1 Partition freihalten als AFS-Partition
- `rpm -i openafs-*.rpm` (Client u.U. weglassen)
- AFS–Serverdienste konfigurieren und starten (Teil 1)
- Client konfigurieren und starten
- AFS–Serverdienste konfigurieren (Teil 2)

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 Zellenname in `/usr/vice/etc/ThisCell` eintragen.

AFS–Serverdienste Teil 1.1

- Datenbankserver starten:

```
> bos create localhost kaserver simple /usr/afs/bin/kaserver \
  -cell clug.de -noauth
> bos create localhost ptserver simple /usr/afs/bin/ptserver \
  -cell clug.de -noauth
> bos create localhost vlserver simple /usr/afs/bin/vlserver \
  -cell clug.de -noauth
```

AFS–Serverdienste Teil 1.2

- Notwendige Einträge in Kerberos–Datenbank erzeugen:
  `> kas -cell clug.de -noauth`

`ka> create afs`	(„Nutzer“ 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 -cell clug.de -noauth`

AFS–Serverdienste Teil 1.3

- Server–Schlüssel hinterlegen
  ```
  > bos addkey localhost -kvno 0 -cell clug.de -noauth
  > bos listkeys localhost -cell clug.de -noauth
  ```

- PTS–Datenbankeintrag für Nutzer „admin“
  ```
  > pts createuser -name admin -cell clug.de -id 1 -noauth
  > pts adduser admin system:administrators -cell clug.de -noauth
  ```

- BOSServer neu starten:
  ```
  > bos restart localhost -all -cell clug.de -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 \
    -cell clug.de -noauth
  ```

- Volume `root.afs` anlegen:
  ```
  > vos create localhost a root.afs -cell clug.de -noauth
  ```

- BOSServer neu starten (ab jetzt wird authentifiziert gearbeitet)
  ```
  > bos shutdown localhost -wait -noauth
  > killall bosserver
  > /usr/afs/bin/bosserver
  ```

AFS–Client konfigurieren

- Zur weiteren Installation wird ein AFS–Client benötigt, da bereits im `/afs`–Verzeichnisbaum operiert werden muss.

- AFS–Client starten:
  `> /etc/init.d/afs start`

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

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 -principal 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
  > ln -s /afs/clug.de /afs/clug
  > fs mkmount /afs/.clug.de root.cell -rw
  ```

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> \
    <toserver> <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 <fileserver> <partition> <volume>`

- Syncronisieren aller RO–Kopien mit dem RW–Volume:
  `> vos release <volume>`

Nutzer anlegen

- Nutzer werden praktisch zweimal registriert: Einmal in der Kerberos–Authentifizierungsdatenbank und einmal in der Protection Database.

- Nutzer anlegen:
  ```
  > kas create <login> -admin <administrationsaccount>
  > pts createuser <login> [<uid>]
  ```

- Es bietet sich an, 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.

Gruppen anlegen

- Die Protection Database 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>`

Referenzen

- http://www.openafs.org/
- http://www.central.org/ - AFS Dokumentation
- http://www.transarc.ibm.com

Nachtrag: Vorführung

Der Vollständigkeit halber hier ein grobes Protokoll der Live–Demonstration.

```
# Anlegen diverser Volumes:
vos create hermes b user
vos create hermes b software
fs mkmount /afs/clug.de/home user
fs mkmount /afs/clug.de/software software
fs copyacl /afs/clug.de /afs/clug.de/home
fs copyacl /afs/clug.de /afs/clug.de/software
# Migrieren eines Volumes:
vos move software hermes b hermes c
vos addsite hemes c root.cell
vos examine root.cell
```

```
vos release root.cell
vos examine root.cell

### dklab2 (2. Fileserver) einrichten ###

# Hier sollte besser der Updateserver verwendet werden!
scp hermes:/usr/afs/etc /usr/afs/etc
bossserver -noauth
# Fileserverprozesse erzeugen
bos create localhost fs fs /usr/afs/bin/fileserver \
  /usr/afs/bin/volserver /usr/afs/bin/salvager \
  -cell clug.de -noauth
bos shutdown localhost -wait
killall bosserver
bosserver
```

```
# wieder auf hermes

# root.{cell,afs} sollte auf jedem Fileserver verfuegbar sein
vos addsite dklab2 a root.cell
vos release root.cell

# Nutzer anlegen

vos create dklab2 a user.chu
fs mkmount /afs/clug.de/home/chu user.chu
kas create chu -admin_user admin
pts createuser chu -id 1000
fs setacl /afs/clug.de/home/chu chu all
useradd -c "Chris" -d /afs/clug.de/home/chu -m -u 1000 chu
```

```
# login als chu

cp /usr/src/packages/RPMS/i386/openafs-* . # Quota reicht nicht
fs listquota /afs/clug.de/home/chu
fs setquota /afs/clug.de/home/chu 500000

# als chu
cp /usr/packages/openafs/* .
# parallel als root/AFS-Admin
fs move user.chu dklab2 a hermes a

# Backup-Volume anlegen und im Homeverzeichnis montieren
vos backup user.chu
fs mkmount BACKUP user.chu.backup
```