AFS at MIT

Basics

What is AFS?

- AFS (now called OpenAFS) is a distributed, caching, authenticated filesystem
- AFS provides a uniform namespace (a file has the same path on any AFS client anywhere)

Components of AFS

- Cache Manager (afsd) - responsible for caching data on the client machine and talking to the servers
- File Server (fs) - Maintains file hierarchy, delivers data, etc
- Protection Server (pts) - maintains access control lists for directories and maps usernames to AFS IDs
- Volume Server (vos) - allows for creation, deletion, and alteration of volumes. Gets its information from the Volume Location Database (vldb) which knows what volumes are stored where

AFS Terminology

- ACL - access control list - a list of entities that are granted access on a specific directory
- Volume - a collection of directories and files that are stored together and can be moved between servers or replicated. Volumes can be assigned quotas to manage resources
- Cell - a collection of volumes. Cells are the first level in the /afs hierarchy. athena.mit.edu is an example of a cell

Volume Naming

- Volumes are named with a prefix according to what the contain:
  - user.jdreed
  - activity.lsc
  - sw.matlab
- To find a volume name, attach the locker, then
  - athena% fs listmount /mit/lockername
  - (or “fs lsm” for short)
  - Result: ‘/mit/consult’ is a symbolic link, leading to a mount point for volume ‘#contrib.consult’.
  - (the ‘#’ in the name is ignored)

Mountpoints

- Volumes can contain mountpoints to other volumes.
- athena% fs lsm /mit/jdreed/OldFiles
  - Result: '/mit/jdreed/OldFiles' is a mount point for volume '#user.jdreed.backup'
- Most volumes have a backup volume that ends in ".backup"

AFS Access Control

- Typical UNIX access controls have 3 “bits” for specifying access - owner, group, and world (everyone else) and 3 modes for each bit (read, write, execute)
- AFS has seven modes (read, list, insert, write, delete, lock, adminster) and up to 16 entities (users, groups, etc) can be on an ACL for a directory
- However, AFS only allows access on a per-directory basis, whereas traditional UNIX permissions allow it on a per-file basis

AFS Tokens

- Tokens identify you to the AFS server.
- Obtained via the “aklog” command from your Kerberos tickets
- You must have tickets to get tokens
- Tokens, once obtained, are no longer related to tickets.
- Tokens can expire while you have valid tickets, and vice versa

Manipulating Tokens

- Get Tokens:
  - athena% aklog
  - athena% renew
- View Tokens:
  - athena% tokens
- Destroy Tokens:
  - athena% unlog
  - (don’t do this if you have any files open)

Manipulating AFS permssions

- athena% fs sa *directory entity modes*
  - *directory* can be “.” or “/mit/joeuser”, etc
  - *entity* can be a username, or a moira group with the prefix “system:” (ie: system:athena-current)
  - *modes* is a combination of the letters r, l, i, d, w, k, a, or the words “read”, “write” or “none”

Special Entities

- system:anyuser
  - Anyone, anywhere, including the web servers
- system:authuser
  - Anyone with an Athena account
- system:expunge
  - The automated nightly expunger, ignore this

Viewing Permissions

- athena% fs la *directory*
  - *directory* is optional, defaults to current dir.
- Results:

```
Access list for . is
Normal rights:
  system:expunge ld
  system:anyuser l
  jdreed rlidwka
```

  - Means that the expunger can list and delete files, any user can list file names (but not contents), and jdreed has full permissions

Using “chmod” in AFS

- People who have used UNIX before will attempt to use “chmod” to set permissions. It won’t work.
- UNIX permissions are used for the “user” bit of the permissions - you can take away the “w” mode, and programs won’t write to it.
- However, anyone with AFS permissions on the directory can restore the “w” mode.

Quotas

- athena% fs lq *directory*
  - Like “fs la”, *directory* is optional
  - Result:

```
Volume Name    Quota    Used  %Used  Partition
user.jdreed  1500000  870472    58%        73%
```

- athena% afsquota *lockername*
  - athena% afsquota joeuser
  - athena% afsquota consult

File Servers

- Sometimes fileservers go down
- To figure out what file server a locker is on:
  - athena% athrun consult lookup *lockername*
- To see if any servers are down:
  - athena% fs checkservers
  - (or just “fs checks” for short)

OldFiles

- “OldFiles” is a copy of the locker as it existed at early this morning (3 or 4 am)
- It does not count towards your quota, and can’t be edited, only read
- It’s actually something called a “mountpoint” for yet another volume

Manipulating Mountpoints

- Make OldFiles temporarily disappear
  - athena% fs rmmount /mit/*username*/OldFiles
- Restore it
  - athena% fs mkmount /mit/*username*/OldFiles user.*username*.backup
- Rename it
  - athena% fs mkm /mit/*username*/backup user.*username*.backup
- The mountpoint must not already exist, and you must not add a trailing “/”.
- “mkm” and “rmm” are short for “mkmount” and “rmmount”