

Hit The Ground Running: AFS

Fifteen minutes of information you
need to understand how to install
and run your own AFS cell

Fastest Possible Overview

- Secure - Kerberos authentication
- Scalable - nothing special needed to grow
- Location independence
 - every client sees same tree
 - users don't know/care about servers
- User control of groups
- Redundancy of static data possible
- Administration from any client system

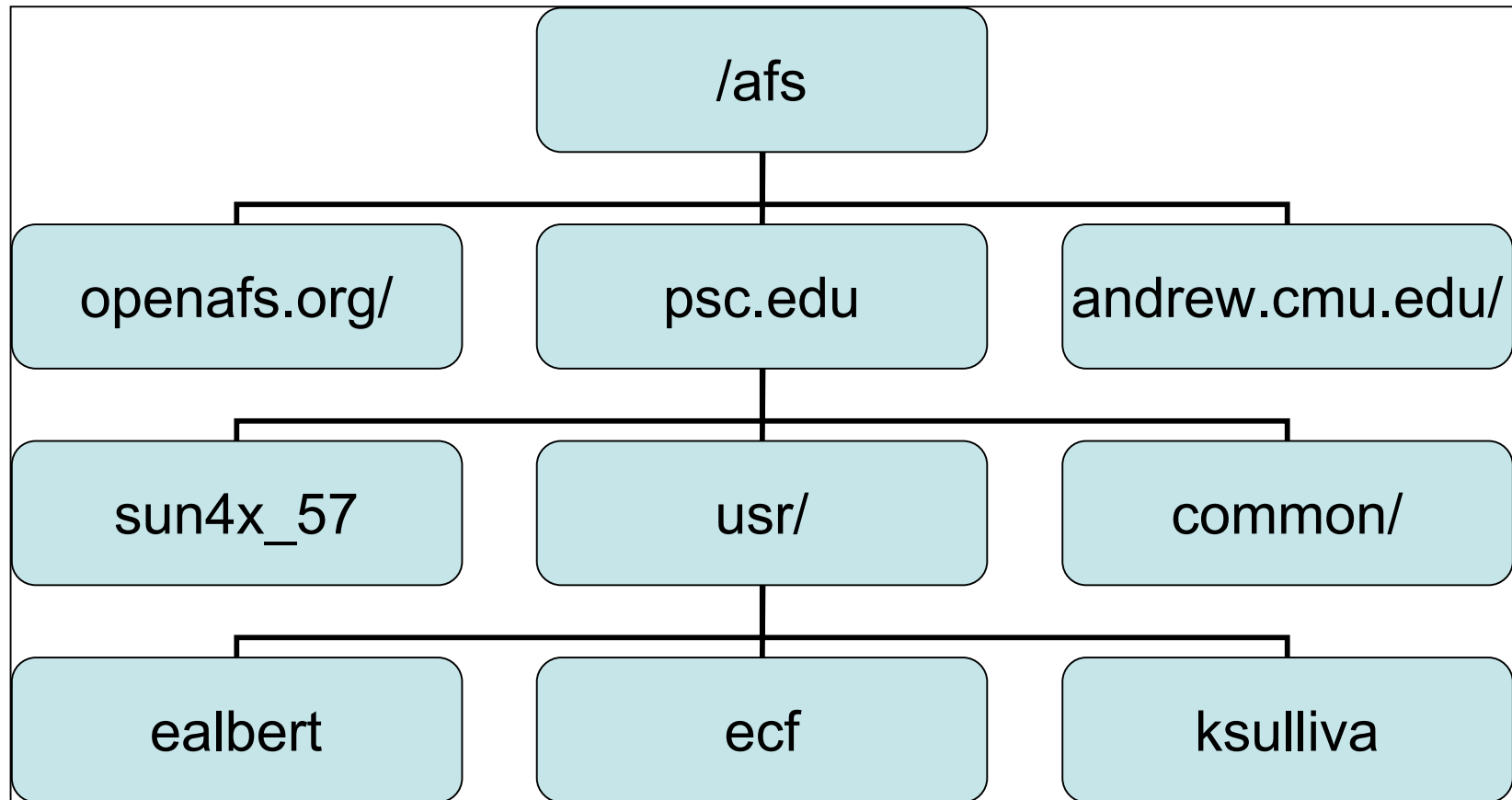
AFS Gotchas

- Can't (yet) do suspend mode for *nix
- Some OSen can't stop & restart client
- No pipes, sockets or device files
- No "byte-range locking"
 - no Oracle dbs. No shared Microsoft files

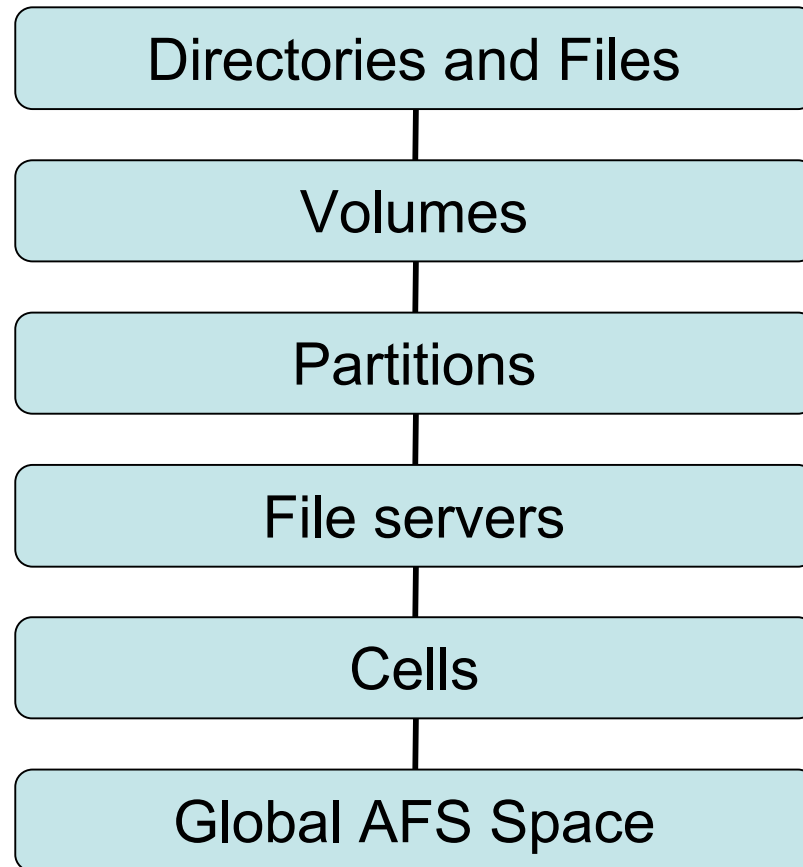
AFS Is Not Unix

- “chown” and “chgrp” require client root **and** AFS administrator privs
- AFS protects directories, not files
 - only the user bits on the unix mode count
- Usage determined via client commands
 - “df” has no use in AFS

What AFS Looks Like (Globally)



Overview of the AFS Universe



Basic Terminology

Cell: One site's AFS setup

- Examples: umich.edu, cern.ch, openafs.org
- Each cell can be made from one or multiple servers
- A University/Company/Organization can have multiple cells

(ie. cmu.edu, cs.cmu.edu, andrew.cmu.edu, sei.cmu.edu)

Basic Terminology

- **Volume:** A collection of files and directories in a separate AFS storage container.
 - **Mount Point** – the point where the AFS volume is placed in the directory structure.
 - Volumes can look like directories
/afs/psc.edu/usr/ecf
- Each of these is a directory *and* a volume *and* a mount point
- Directories are not always volumes
/afs/psc.edu/usr/ecf/private
- “private” is a directory within the volume for “ecf”

Volumes & Quota

- Each volume has it's own quota
- A full volume does not affect other volumes around it or on the same server
- Determine quota with either

- **fs quota**

85% of quota used

- **fs listquota** (or **fs lq**)

Volume Name	Quota	Used	%Used	Partition
usr.2.ealbert	500000	422822	85%	71%

The Cache

- **Cache:** The space on the local disk where AFS stages files between the server and showing them to you.
 - Stores pieces of files, to allow faster access of recently viewed files
 - Works to help make sure clean data is written back to the server
 - Keeps track of where recently viewed files are both in cache and on servers

The Cachemanager

- Also known as “**afsd**”, the processes that talk to the servers and manage the cache
- You’ll notice multiple ones running (on *nix boxes)
- Very kernel intensive, which is why there are clients for limited OSes

Authentication

- Kerberos or Active Directory
- Not currently shipping with Kerberos installation, but hooks are there
- Encryption on both sides (client & server), nothing in the clear
- Kerberos 5 (VERY) strongly encouraged
 - AD, MIT or Heimdal, your pick

AFS Command Suites

- **fs** - controls local client and cache manager, also sets quota and privs on volumes - requires root and/or *admin* privs as needed
- **pts** - controls protection db, modifying users and groups - most commands not privileged
- **vos** - volume manipulation - most commands require *admin* and fileserver admin privs
- **backup** - controls the backup server
- **bos** - AFS server controls - except for “status” all commands require privs.

A Few Words About Groups

- **pts** allows users to create their own groups
- Users can use multiple groups for protecting different directories
- Admins can create special “self-owned” groups so more than one person can own and control a group and it’s sub-groups
 - Useful for projects that involve sharing lots of directories of data

RLIWDKA

- R: read files
- L: lookup, or list files [ability to ls]
- I: insert file [write it if it doesn't already exist]
- W: write, or modify
- D: Delete
- K: Lock [advisory lock]
- A: Administer, or change the protections in this directory

AFS Servers

- Server software for all client OSes **and** FreeBSD and NetBSD.
- **DO NOT RUN WINDOWS SERVER.** Hasn't been worked on since the dawn of man.
- Fileservers tend to be *very* I/O bound
- Decent hardware but don't have to bleed
 - we use RAID 5, paying the price of speed for stability

AFS Server Processes

- **Bosserver** - Starts and monitors all processes, restarts if they die, can do cron-like changes
- **Fileserver** - passes files back and forth with the Cache Manager, monitors changes by the “fs command”
- **Volserver** - handles volume manipulation: creation/deletion, movement, cloning and backups
- **Salvager** - performs consistency checks and repairs on volumes

These make up the basic “AFS server”

AFS DB Server Procs

- **vlserver** - volume location server, keeps track of all volumes & maintains a db
 - **ptserver** - protection server - maintains user access and groups
 - **buserver** - optional backup server
 - **[kserver]** - don't.
-
- These run in addition to previous processes
 - DB servers don't have to serve files (but often do)

DB Servers & Ubik

- If running K5 can put KDCs on DB servers
- Minimum of 1 DB server, Max suggested at 5
 - more than 5 and things can get bogged down
 - 3 is a nice number, depends on size of your cell
- **Ubik** keeps databases in sync
 - servers vote on master (“sync”) site
 - in case of even numbers, lowest IP gets 2 votes

Read Only Clones

- adds redundant availability for *static* data
 - not good for user volumes or other things that change regularly
- generally clones are created on demand
- if one clone becomes unavailable, client will automatically switch to another
 - however if all RO clones are unavailable, RW will not be used unless specifically requested

Backups & “OldFiles”

- AFS can create a nightly backup of each volume
- Reduces the need to ask for a file restore!
- It is read-only
 - You cannot change it
 - You *can* copy files from it
 - It does not affect any other volume’s quota

For More Information

- www.openafs.org - OpenAFS web site
- www.stacken.kth.se/projekt/arla - Arla web site
- AFS Guru session - Thursday 9 am
Royal Palms Salon 6
- Talk to me in the Hallway Track
- This talk:
<http://www.pmw.org/~ecf/afs/>