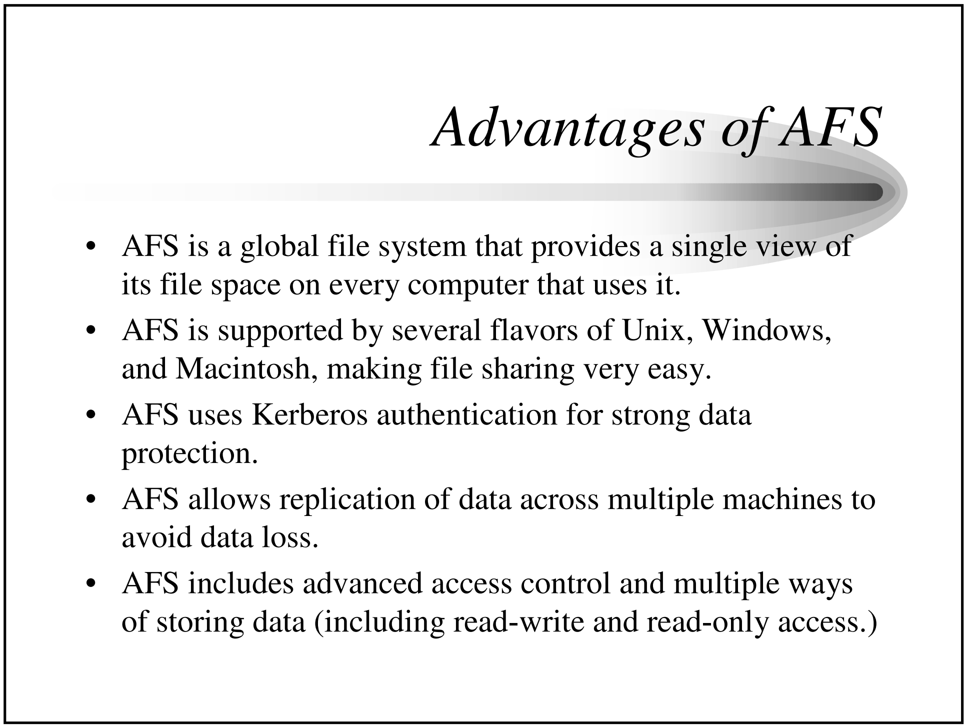
Introduction to AFS
IMSA Intersession 2003

An Overview of AFS

Brian Sebby, IMSA ’96

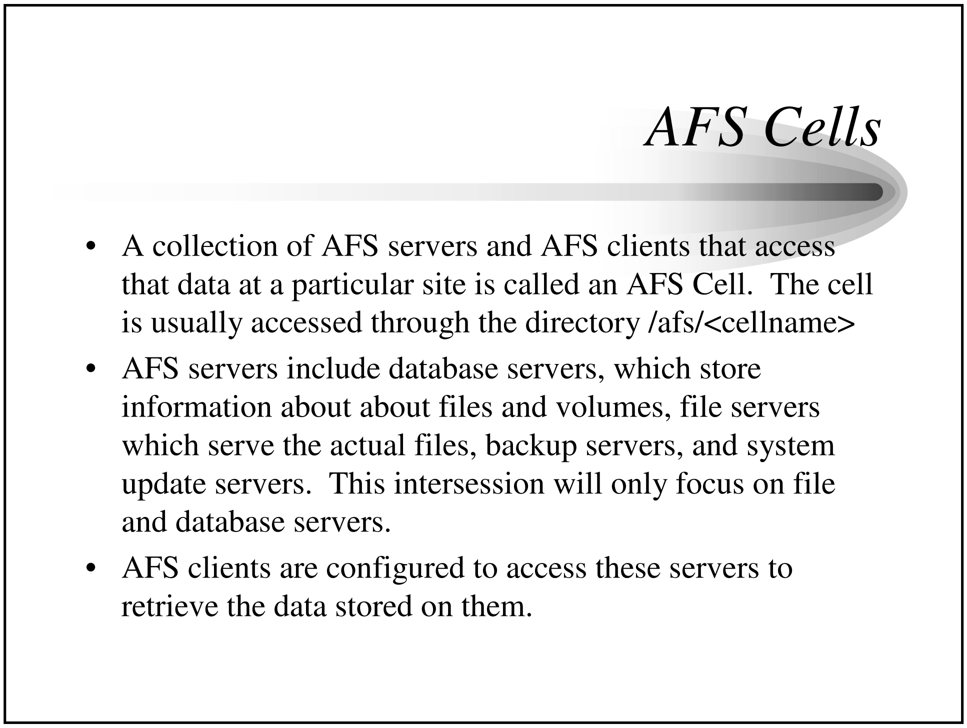
A Brief History of AFS

- Started as research project at Carnegie-Mellon University in the 1980s.
- AFS group created Transarc Labs to develop AFS as a commercial product in the late 80s.
- Transarc Labs acquired by IBM in mid-90s.
- AFS source code released to open source community in November 2000, OpenAFS project begun.
- IBM/Transarc announces end-of-life of their commercial version of AFS in mid-2002.
- OpenAFS project continues, version 1.2.8 released in December 2002.



Advantages of AFS

- AFS is a global file system that provides a single view of its file space on every computer that uses it.
- AFS is supported by several flavors of Unix, Windows, and Macintosh, making file sharing very easy.
- AFS uses Kerberos authentication for strong data protection.
- AFS allows replication of data across multiple machines to avoid data loss.
- AFS includes advanced access control and multiple ways of storing data (including read-write and read-only access.)



AFS Cells

- A collection of AFS servers and AFS clients that access that data at a particular site is called an AFS Cell. The cell is usually accessed through the directory /afs/<cellname>
- AFS servers include database servers, which store information about about files and volumes, file servers which serve the actual files, backup servers, and system update servers. This intersession will only focus on file and database servers.
- AFS clients are configured to access these servers to retrieve the data stored on them.

AFS Volumes

- Files are stored in AFS in logical units called volumes. Volumes can contain many subdirectories and files.
- Volumes are linked to each other through mount points that function like regular directories.
- There are three types of volumes: read-write, read-only, and backup volumes.
- Quotas in AFS are handled on a per-volume basis, and apply to every user of that volume.

More about volumes

- When a new volume is created, it is by default a read-write volume. All access to this volume is initially to the RW volume.
- If a volume is replicated, the replicas are all read-only volumes.
- Once the volume has been replicated, further accesses are through the read-only volumes.
- Mount points can be read-only or read-write. The only way to access a read-write volume is if its parent mount points are read-write volumes.

Even more about volumes

- When you access /afs/<cellname>/<some volume> you normally will be accessing the read-only volume if one exists.
- It is common to create a read-write mount point at /afs/.<cellname> to allow access to read-write volumes. Any access to a volume under this read-write mount point will be in read-write mode.
- Mount points do not contain info about the volume's location, that is handled by the database server.
- A backup volume is a snapshot taken of a volume at a particular time. It cannot be modified. Backup volumes are used to provide quick access to older data (usually yesterday's data) and for backing up the AFS file system.

Tokens and ACLs

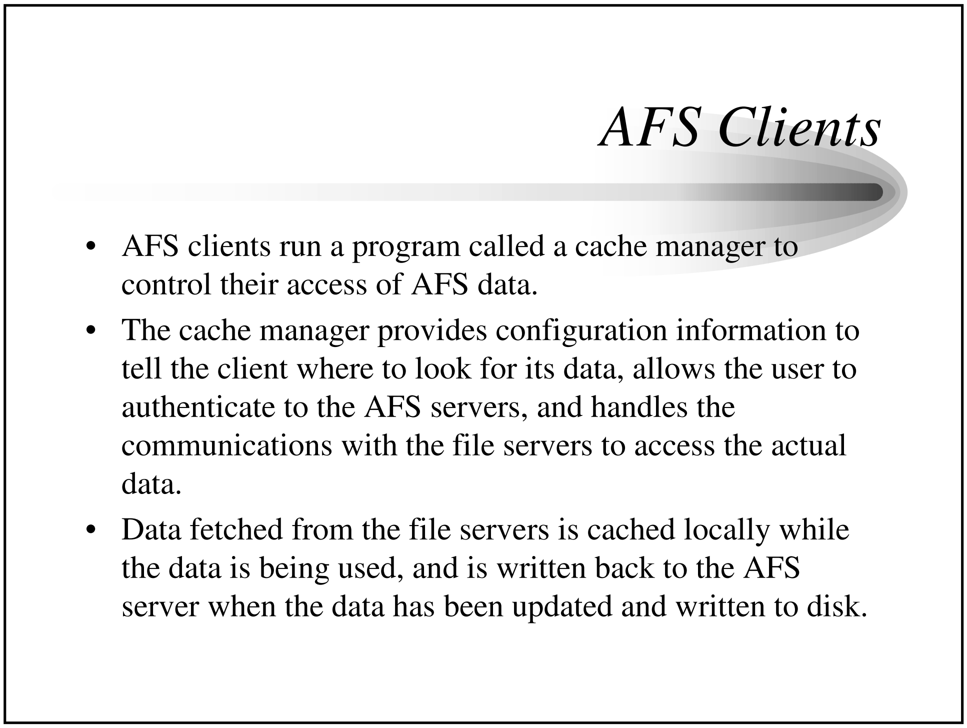
- When a user authenticates to AFS, they are given a token that lets the servers know what privileges that user has.
- The AFS token is really a Kerberos v4 ticket. Kerberos v5 tickets can be converted to be used with AFS, but we will not be covering how to do so in this intersession.
- A user can have only one token per cell that they are authenticated to.
- A token can be associated either with a user's UID or a protection unit called a PAG that is unique to a shell process.
- Access Control Lists (ACLs) are used to allow access to AFS data. Unlike normal Unix protection bits, ACLs are set on a directory basis and apply to every file in that directory.

AFS Servers

- AFS servers provide database and file server access.
- The database servers have four databases that they manage: the authentication, protection, volume location, and backup databases.
- The authentication database stores account passwords, and the server's encryption key.
- The protection databases stores user names and UIDs, group names and GIDs, and group members.
- The volume location database stores the physical location and IDs of all volumes in the cell.
- The backup database stores the information about backups of the AFS data.

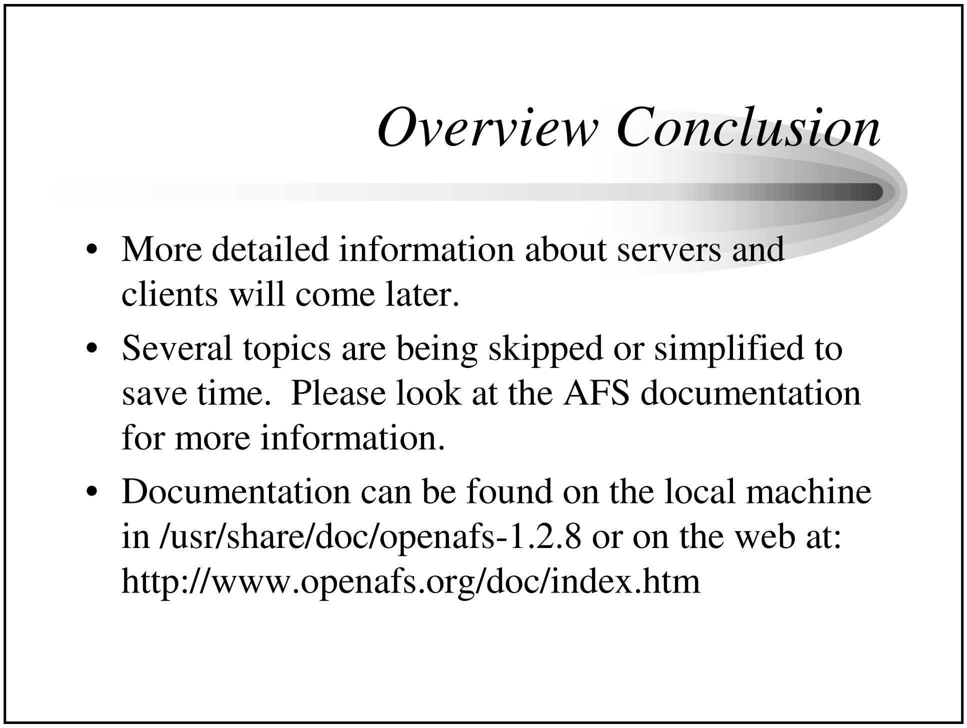
More about servers

- The file servers handle the physical access to AFS data. They store the volumes on their disks, and provide the data when requested.
- The file servers store an ACL for each directory in the file space, and verify that users have the authority to access the requested data based on the user's token.
- Volumes are stored in "vice" partitions on fileservers that are mounted as /vicep<a-z>.
- There are also system control servers and binary distribution servers which can distribute configuration and AFS binaries to other servers and clients, but we will not cover these.



AFS Clients

- AFS clients run a program called a cache manager to control their access of AFS data.
- The cache manager provides configuration information to tell the client where to look for its data, allows the user to authenticate to the AFS servers, and handles the communications with the file servers to access the actual data.
- Data fetched from the file servers is cached locally while the data is being used, and is written back to the AFS server when the data has been updated and written to disk.



Overview Conclusion

- More detailed information about servers and clients will come later.
- Several topics are being skipped or simplified to save time. Please look at the AFS documentation for more information.
- Documentation can be found on the local machine in /usr/share/doc/openafs-1.2.8 or on the web at: http://www.openafs.org/doc/index.htm