Case Study:

Andrew File System

What is AFS ?

- AFS is a distributed file system that enables file sharing across both local area and wide area networks.

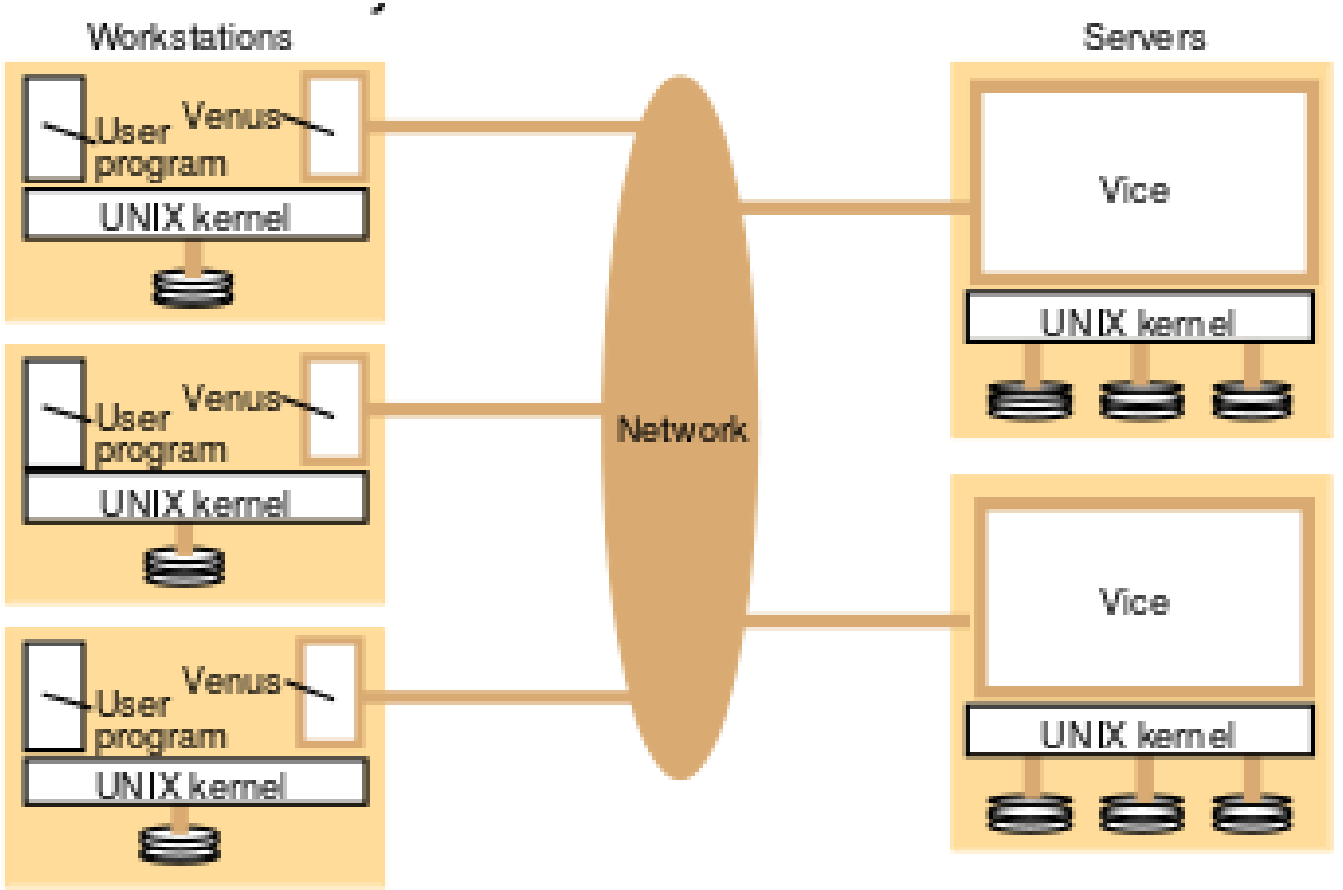
- It was developed by **Carnegie Mellon University** as part of the **Andrew Project**.

Design decisions for AFS

- **whole-file serving** : entire contents of directories and files transfered from server to client.

- **whole file caching**: when file transfered to client it will be stored on that client’s local disk

AFS Architecture : Venus and Vice



Callback mechanism

- ensures that cached copies of files are updated when another client performs a close operation on that file
- **callback promise**
  - token stored with cached file
  - **status: valid or cancelled**
- when server performs request to update file (e.g., following a close), then it sends **callback** to all Venus processes to which it has sent **callback promise**
  - RPC from server to Venus process
  - Venus process sets **callback promise** for local copy to cancelled
- Venus handling an open
  - check whether local copy of file has valid **callback promise**
  - if canceled, fresh copy must be fetched from Vice server

Callbacks and Callback Promises

User process	*UNIX kernel*	*Venus*	*Net*	*Vice*
open(FileName, mode)	If *FileName* refers to a file in shared file space pass the request to Venus. Open the local file and return the file descriptor to the application.	Check list of files in local cache. If not present or there is no valid ***callback promise*** send a request for the file to the Vice server that is custodian of the volume containing the file. Place the copy of the file in the local file system, enter its local name in the local cache list and return the local name to UNIX.	→ ←	Transfer a copy of the file and a ***callback promise*** to the workstation. Log the callback promise.
read(FileDescriptor, Buffer, length)	Perform a normal UNIX read operation on the local copy.			
write(FileDescriptor, Buffer, length)	Perform a normal UNIX write operation on the local copy.			
close(FileDescriptor)	Close the local copy and notify Venus that the file has been closed	If the local copy has been changed, send a copy to the Vice server that is the custodian of the file.	→	Replace the file contents and send a ***callback*** to all other clients holding ***callback promises*** on the file.

Cache Consistency and Concurrency Control

- AFS does not control concurrent updates of files, this is left up to the application

- cache consistency only on open and close operations

Security

- AFS makes use of Kerberos to authenticate users
- AFS uses access control lists(ACL) to restrict access to file directories