Active Disk File System

A distributed scalable file system

Hyeran Lim,
Vikram Kapoor,
Chirag Wighe,
Prof. David H.-C. Du
Dept. of Electrical Engineering and Computer Science

University of Minnesota

Active disk file system

- Introduction - Motivation, technologies and review
- Advantages and challenges
- Programming model
- Conclusions and future work

Active Disks evolution

- Increasing processor speeds
- Decreasing memory footprints
- Decreasing costs
- Evolution of drives
  - Early disk drives
  - SCSI, IDE drives
  - Network Attached storage Disks
  - Active disks

Advantages of Active Disks

- Removes File System from Critical Path
  - Transfer Data directly between Clients and disks
  - Offload functionality of File System to disks
  - Number of disks compensate for slower MIPS
  - Example: Self Management of Disk Space

Advantages of Active Disks (Cont...)

- Executing Application Specific code on Disk
  - To use idle cycles present at disk!
  - It still takes milliseconds to read a block from disk
  - Suitable for filtering, storage management, specialized support

Challenges

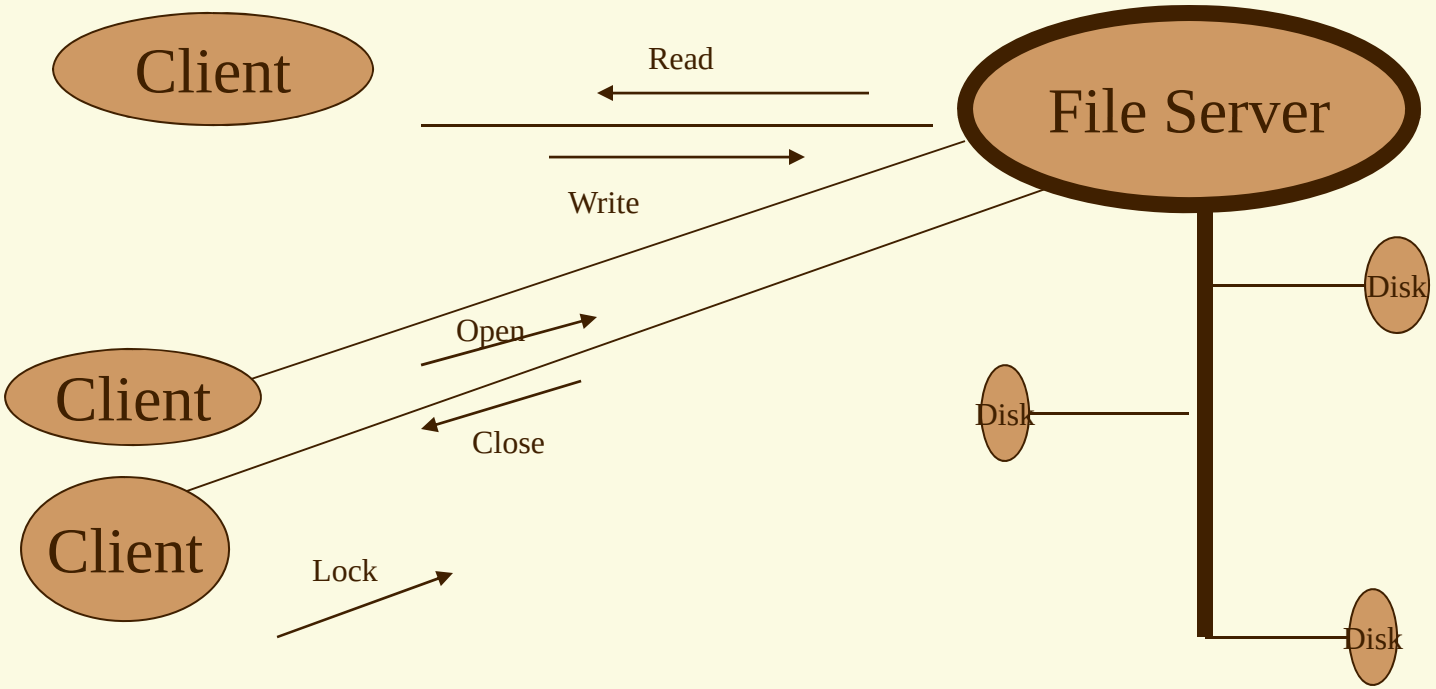
- File System
  - Exploit processing power at disk
  - Scalable
  - Support transparent replication
  - Dynamic components
  - Security
  - Minimize work at File Manager

Applications

- Distributing user computation
  - e.g. database select
  - attacks I/O bus and network bandwidth bottlenecks
- Distributing system computation and state
  - distributing file system functionality
  - horizontal state distribution

File system perspective on the evolution of storage hardware

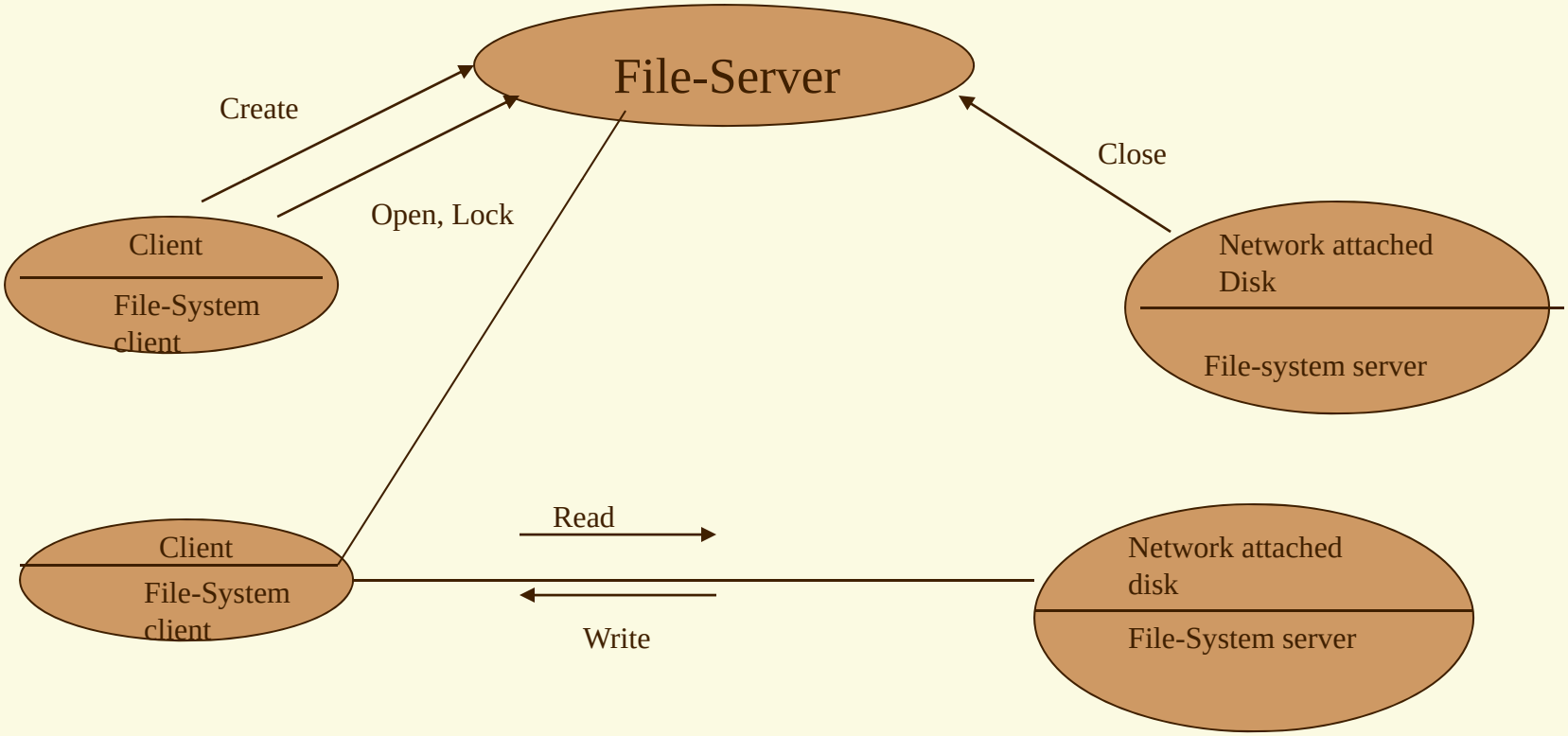
Centralized server file-system NFS



Central Server e.g. NFS file server

- All remote requests to server
- Central server bottleneck
  - Server I/O bus
  - Server processor
  - Server memory

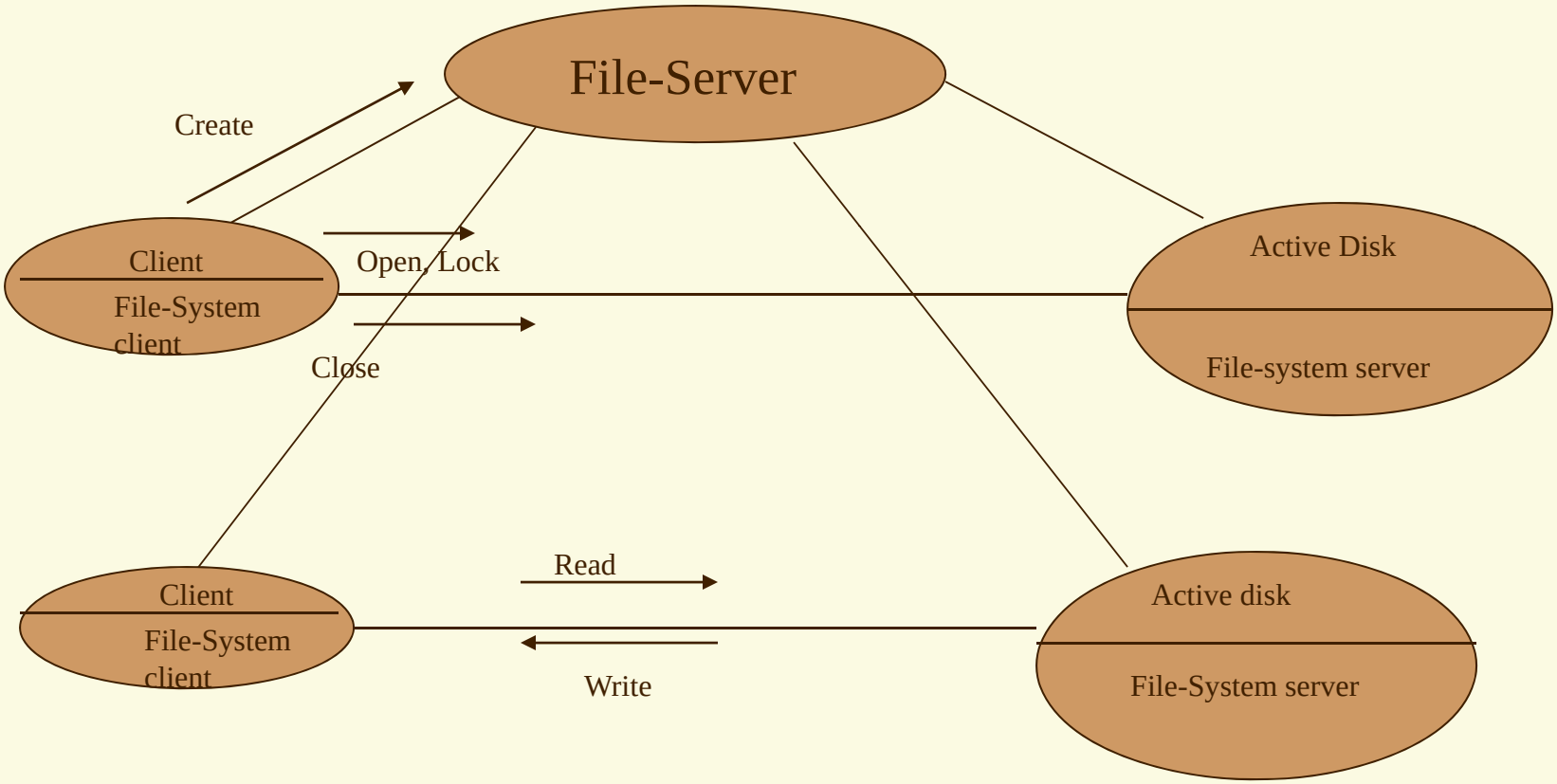
Network Attached Storage file-system



Network attached storage File System

- Lookup at central file server
- Read, Write at network attached disks
- Reduces load on central server I/O bus
- Bottlenecks
  - State information
  - Network links at server

Active disk file system

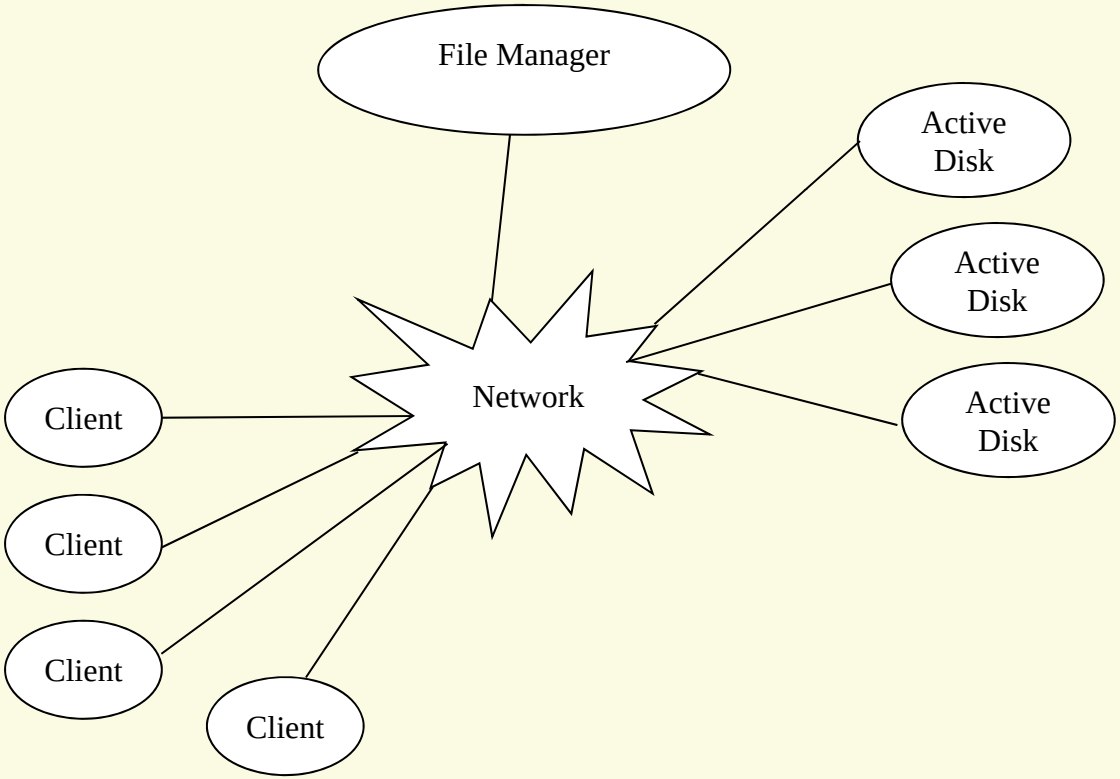


Active disk file system

- Read/Write to active disk
- State information at active disk
- Open/Close/Lock to active disk
- Create etc at file manager

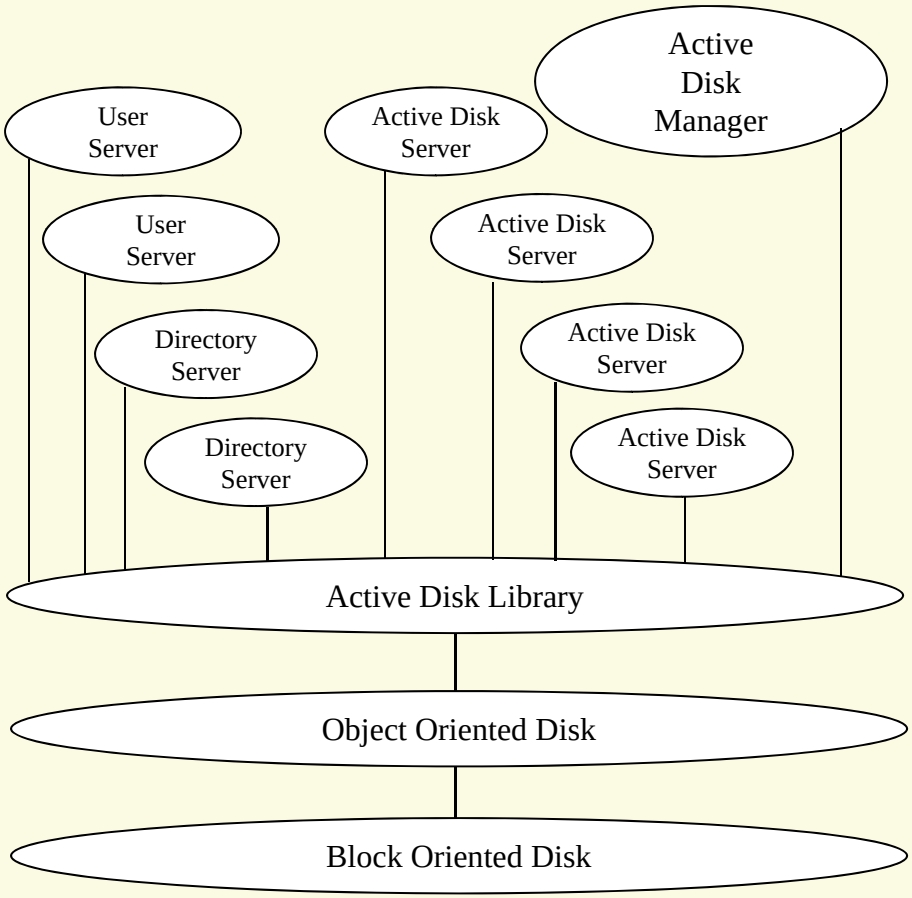
Design and Implementation

Active disk system architecture



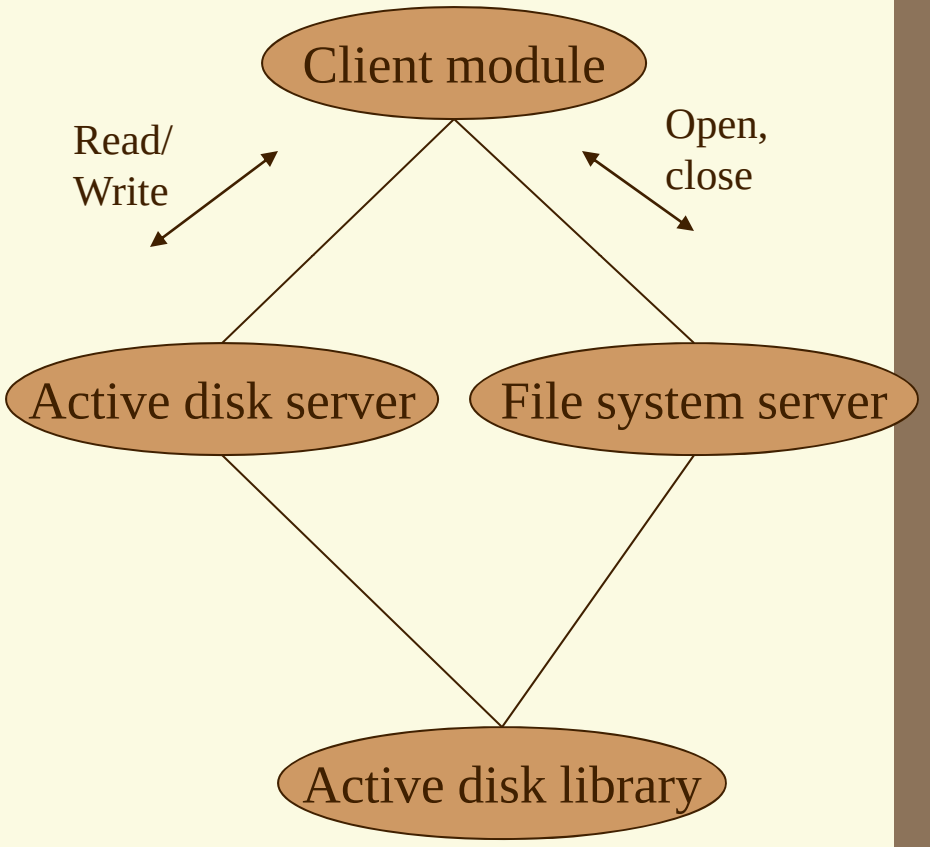
Active Disk File System

Active Disk

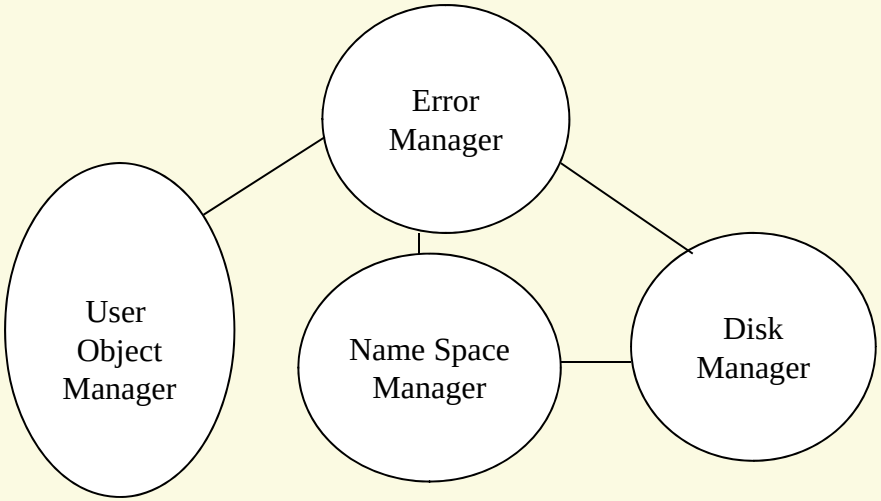


Active Disks Cont..

- Server modules
  - Active disk server
  - file system server
  - Active disk manager
- File system interface



File Manager



Design - Cont..

- Client
  - Active disk client
  - File system client
- File Manager
  - Disk manager
  - central file server
    - create
    - redistribution

Implementation

- Lookup
  - sever semantics - parse as much as possible
  - requires directory knowledge
  - eliminates pathname recursion
  - client semantics

Implementation - Cont.

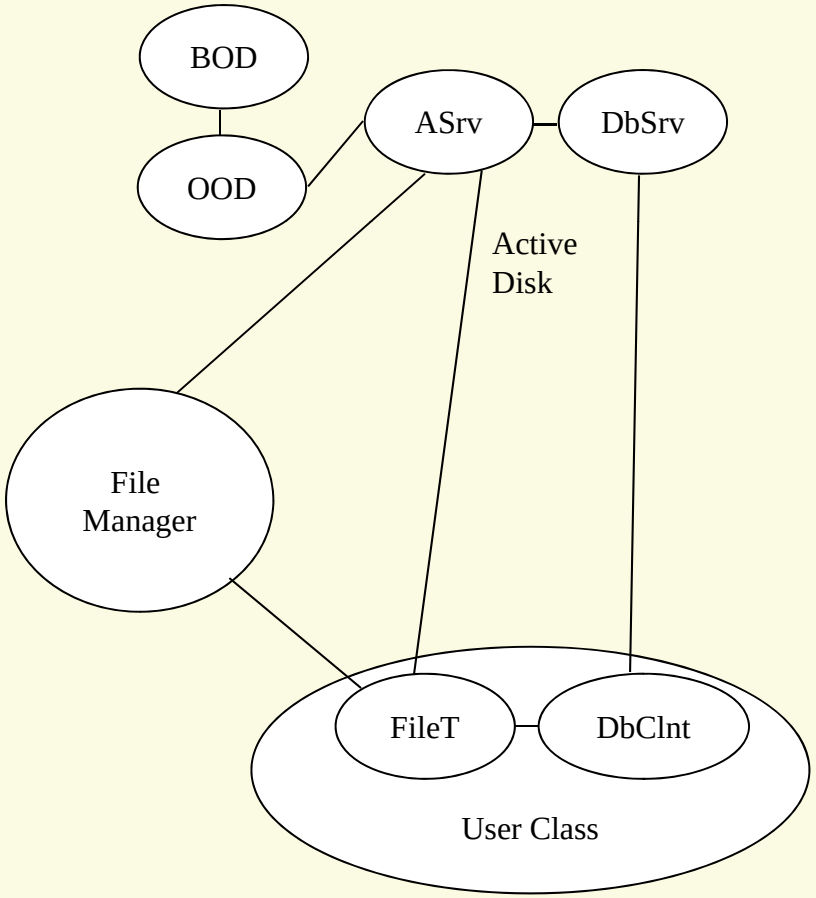
- File system server (directory)
  - searching directories
  - creation of objects
  - deletion of objects
- File system client (directory)
  - interaction with file manager
  - creating / removing objects

Implementation - Cont.

- Stateful file system
  - Stateful file service
    - state information maintained at active disk
  - Open file / lock tables at disk

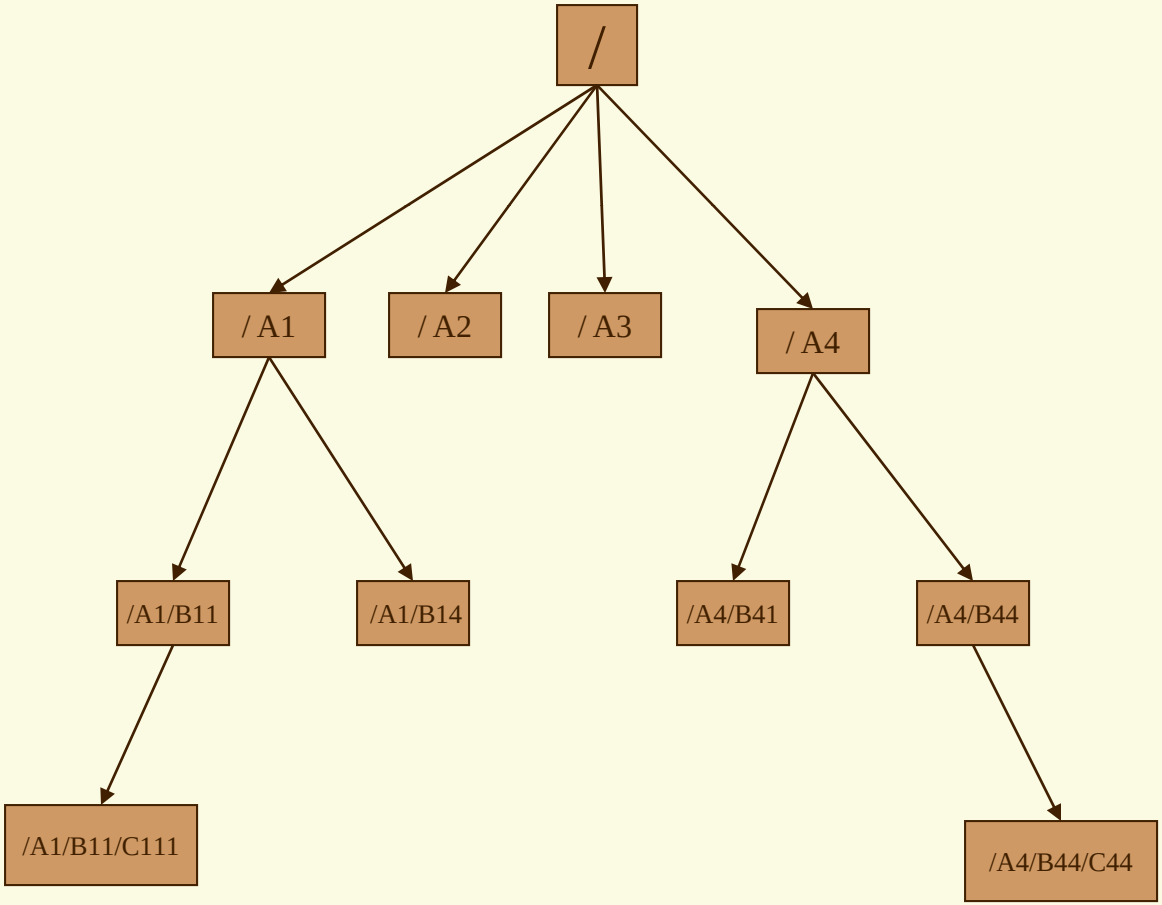
Implementation (Cont....)

Database Select



Proof of concept

Sample file system



Pathname recursion

64 request / sec. 1 Kbytes directories

Type of file system	Maxm. Load on I/O bus of server or network disk or active disk in Kbytes/Sec	Network bandwidth used Kbytes/Sec	Average latency per lookup request in terms of RPC calls
NFS(Centralized file server)	128	Negligible	1
NASD based file system	32	128	3
ADFS – Optimal case	32	Negilgible	1
ADFS – Worst case (overloaded disk)	128	Negligible	1
ADFS –Worst case(Cross referenced directories)	32	128	3

Conclusions - Advantages

- Pathname recursion
  - reduces I/O bus, network bandwidth
  - reduces latency(RPC calls)
- Distributing state information
  - reduces processor/memory load on server
- Active disk storage
  - reduces I/O bus load on server

Advantages - Cont.

- Offload user level computation
  - reduce network bandwidth
  - e.g. database select
- Offload file manager functions
  - naming most prominent file server function
  - No file manager?
  - Fault tolerance / availability

Limitations in design

- Distribution of directory structure
  - Should utilize pathname recursion
- Distributing requests across active disks
  - avoid overloading a few active disks
- Capabilities of active disks?
  - How much to put there?
  - Compromise possible

Limitations in Implementation

- Use actual hardware
  - use high speed switching fabric
- Load file system based on real life data
- Run commonly used applications
  - offloaded functionality

Future Work

- Fault Tolerance and Replication
- Error Management
- Caching
- Dynamic behavior
- Load Balancing and managing user servers
- Security
- Processing requirements on the Active disk

Selected references

- Erik Riedel, Garth A. Gibson, Christos Faloutsos. Active Storage For Large-Scale Data Mining and Multimedia, (VLDB '98)
- Seagate Technology, Inc. white paper. Object Oriented Devices: Description of requirements
- Garth A. Gibson, David F. Nagle, Khalil Amiri, Fay W. Chang, Eugene Feinberg, Howard Gobioff, Chen Lee, Berend Ozceri, Erik Riedel, and David Rochberg. A Case for Network-Attached Secure Disks, Technical Report CMU-CS-96-142

References - Cont.

- Garth A. Gibson, Dave F. Nagle, Khalil Amiri, Jeff Butler, Fay W. Chang, Howard Gobioff, Charles Hardin, Erik Riedel, David Rochberg, Jim Zelenka. Filesystems for Network-Attached Secure Disks, CMU SCS technical report CMU-CS-97-118, 1997.