ĐẠI HỌC

BÁCH KHOA

25 YEARS ANNIVERSARY

SOICT

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Chapter 2
File management

Overview

- Filenames
  - File Identity
- Directories (folders)
  - Group of files in separate collections
- Metadata
  - Creation time, last access time, last modification time
  - Security information (Owner, Group owner)
  - Mapping file to its physical location of file (e.g. location in storage devices)
- Computer file
  - A resource for storing information
  - Durable, remained available for access
  - Data: sequences of bits
- File system
  - Control how computer file are stored and retrieved
  - Main operators: READ, WRITE (offset, size), CREATE, DELETE

Local vs. distributed file systems

Local file systems

NTFS

Ext4
File System

hadoop
HDFS

NFS
NETWORK FILE SYSTEM

The Google File System

By Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
(Presented at SOSP 2003)

PVFS

Distributed file system

- File system
  - Abstraction of storage devices
- Distributed file system
  - Available to remote processes in distributed systems
- Benefits
  - File sharing
  - Uniform view of system from different clients
  - Centralized administration

Goals: Network (Access) Transparency

- Network (Access) Transparency
  - Users should be able to access files over a network as easily as if the files were stored locally.
  - Users should not have to know the physical location of a file to access it.
- Transparency can be addressed through naming and file mounting mechanisms
  - Location Transparency: file name doesn’t specify physical location
  - Location Independence: files can be moved to new physical location, no need to change references to them. (A name is independent of its addresses)
  - Location independence → location transparency, but the reverse is not necessarily true.

Goals: Availability

- Availability: files should be easily and quickly accessible.
- The number of users, system failures, or other consequences of distribution shouldn’t compromise the availability.
- Addressed mainly through replication.

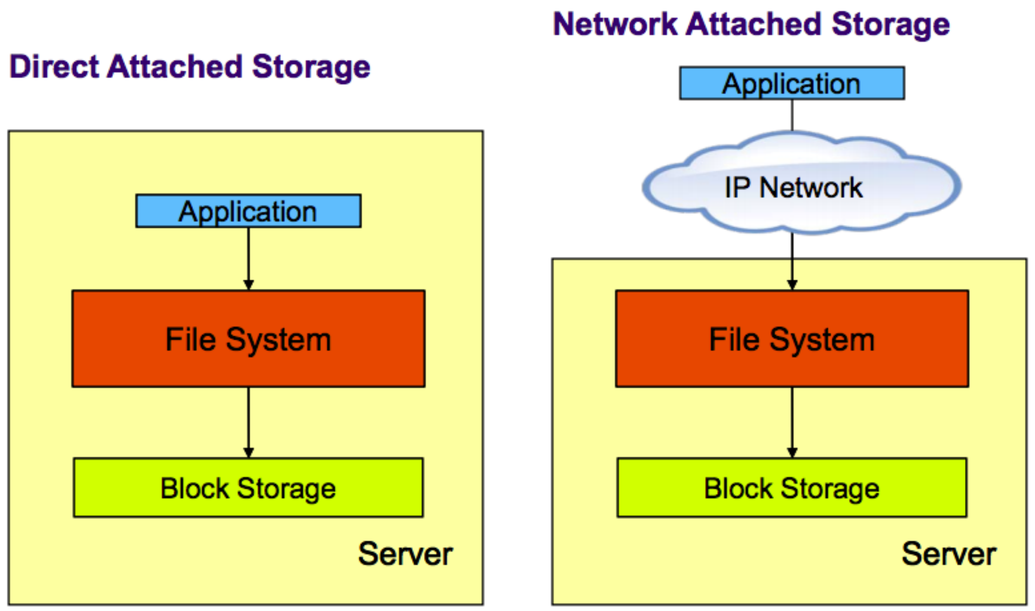
Architectures

- Client-Server
  - Sun Microsystem Network File System (NFS), Google File System (GFS)
  - Architecture
    - One or more machines (file servers) manage the file system.
    - Files are stored on disks at the servers
    - Requests for file operations are made from clients to the servers.
    - Client-server systems centralize storage and management; P2P systems decentralize it.
- Symmetric
  - Fully decentralized; based on peer-to-peer technology
  - e.g., Ivy (uses a Chord DHT approach)

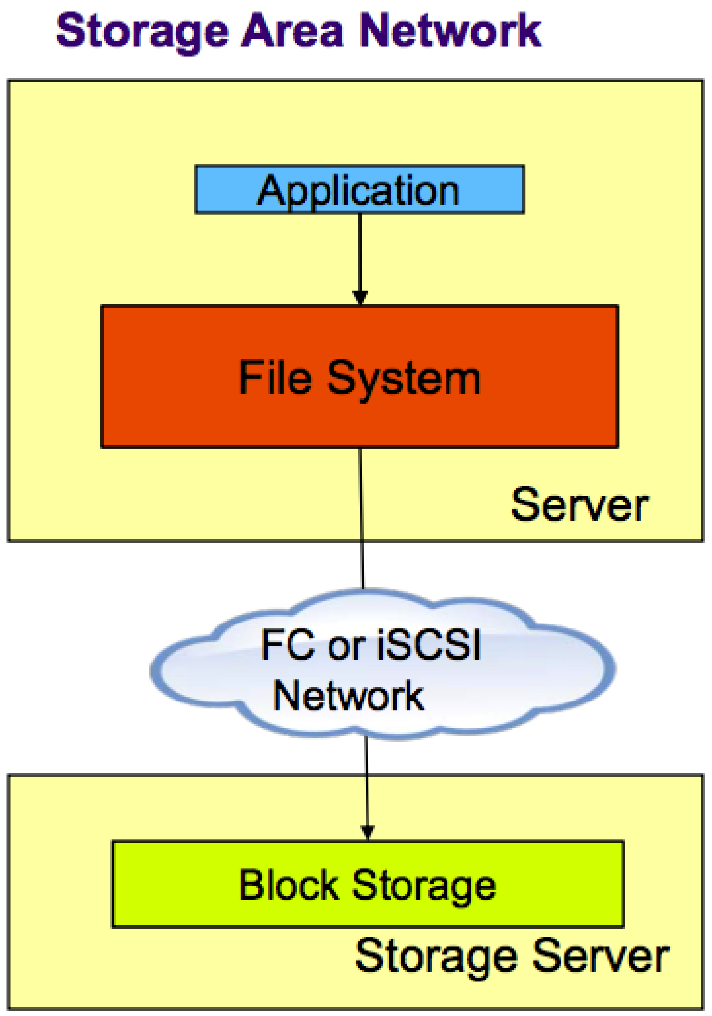
The Evolution of Storage

- Direct attached storage (DAS)
- Network attached storage (NAS)
- Storage area network (SAN)
- Combined technologies

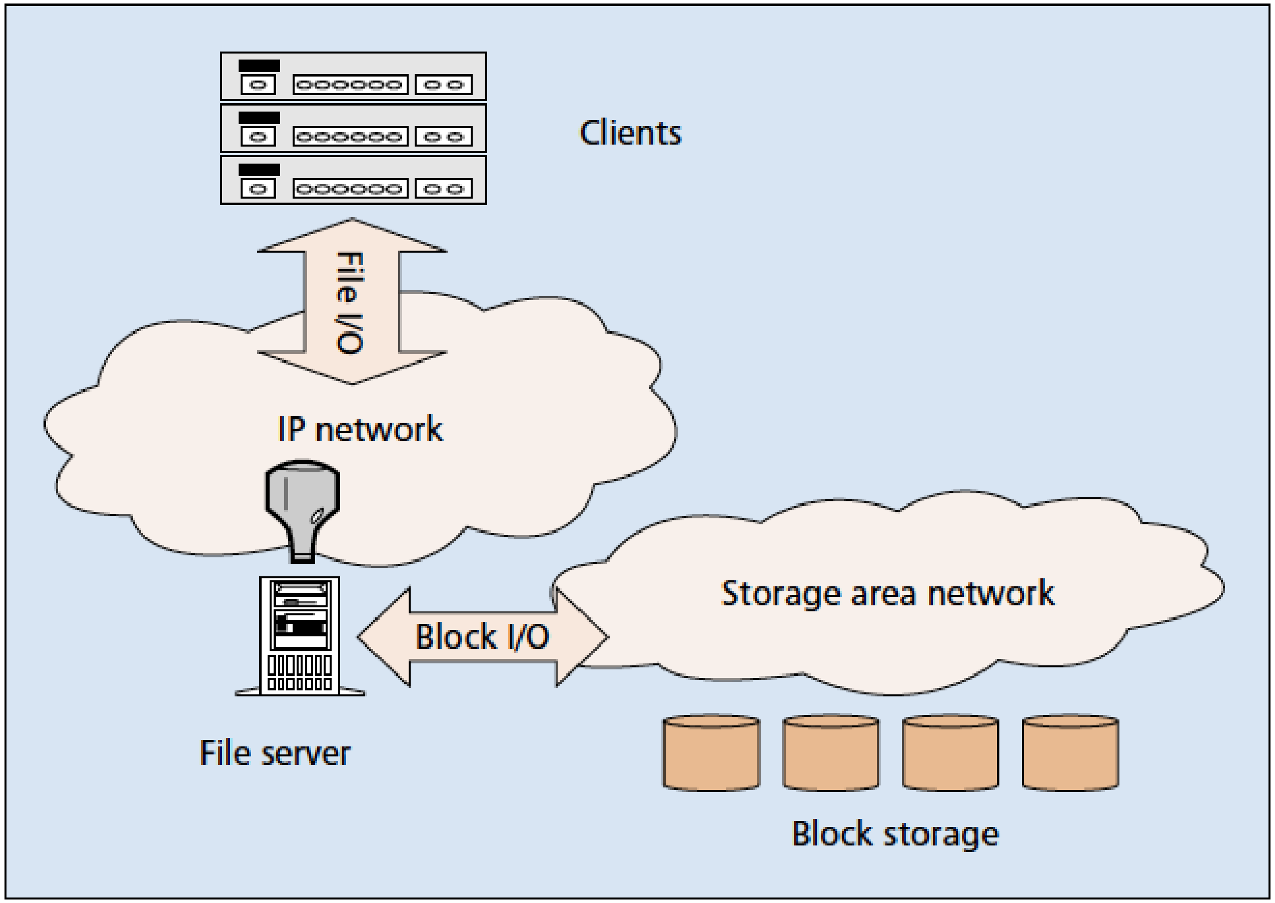
From DAS to NAS



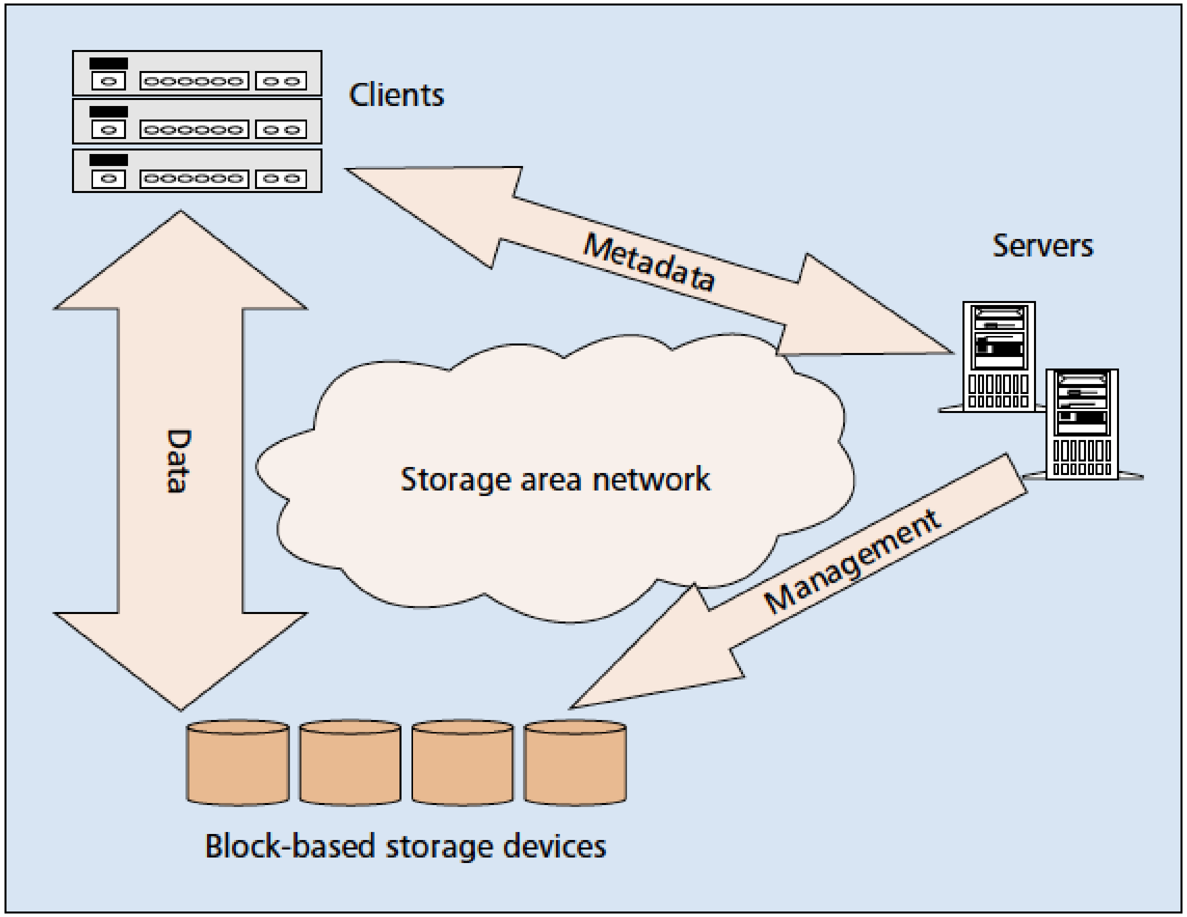
To SAN

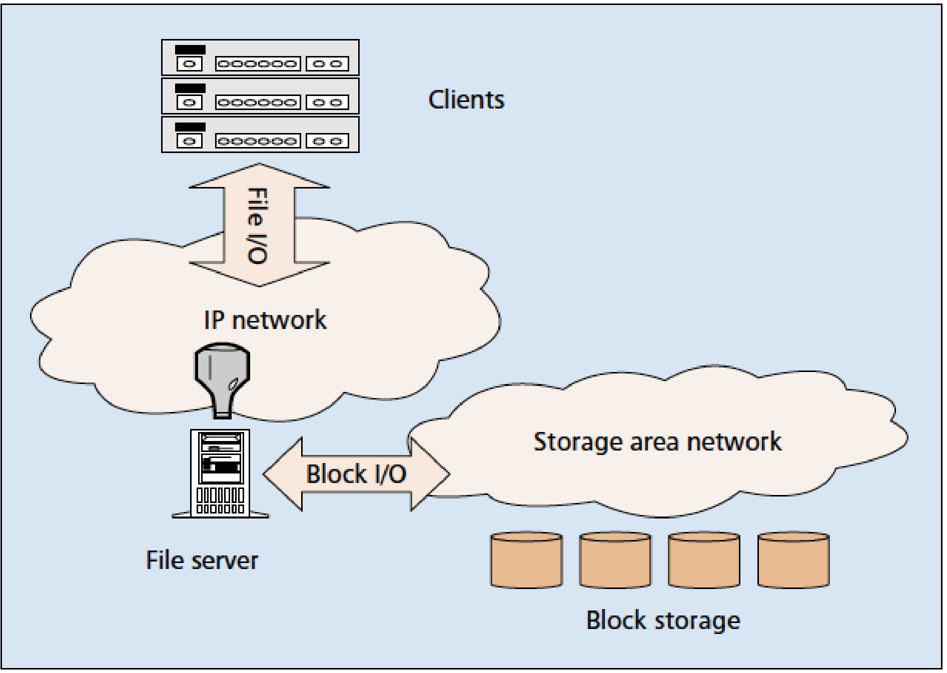

Storage Area Network
Application
File System
Server
FC or iSCSI Network
Block Storage
Storage Server

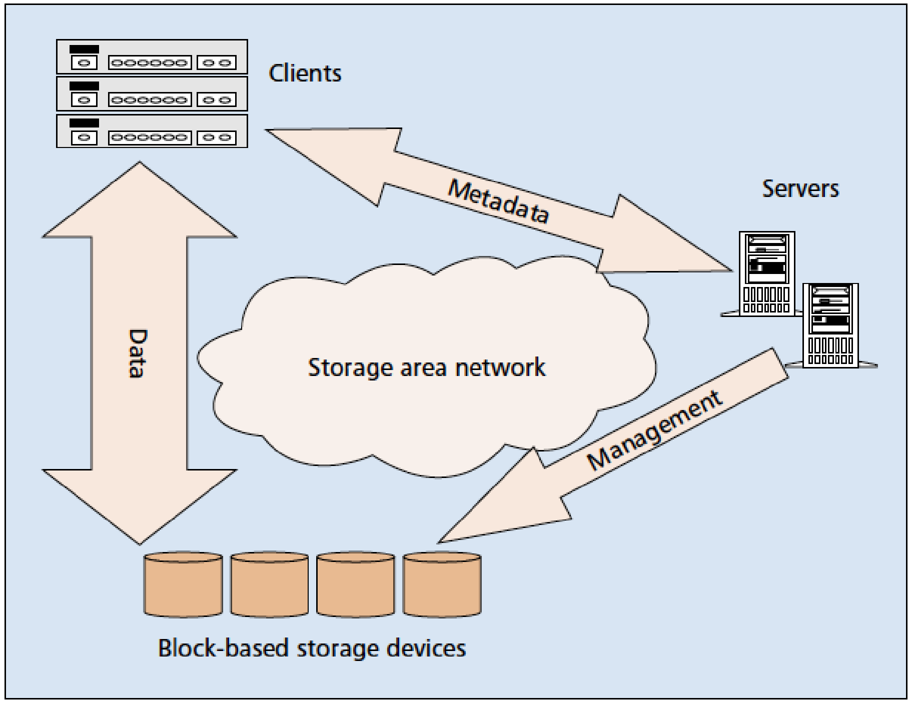
NAS file system



SAN file system



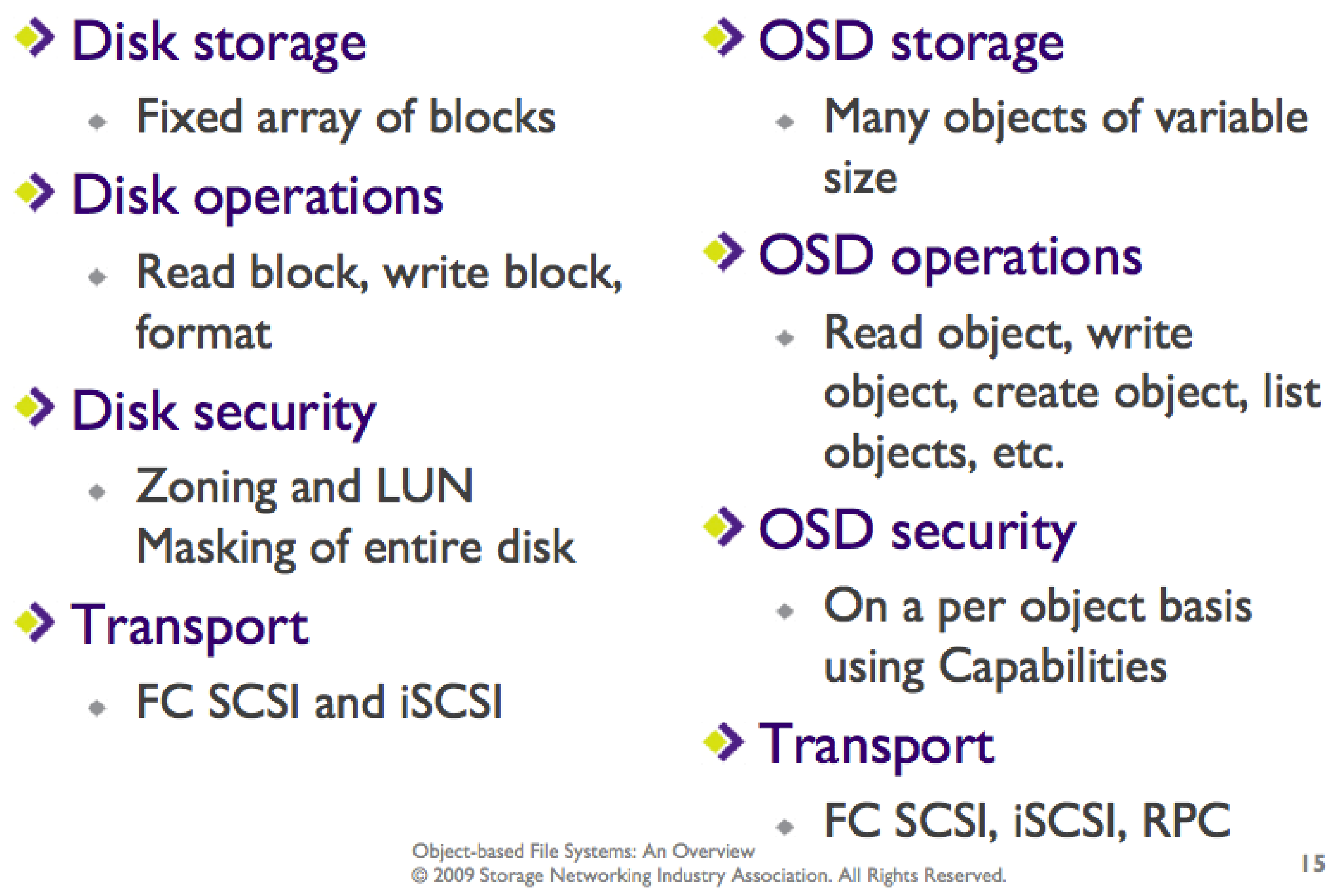

Clients
File I/O
IP network
Block I/O
Storage area network
File server
Block storage


Clients
Metadata
Servers
Data
Storage area network
Management
Block-based storage devices

Object storage device (OSD)

- OSDs hold objects rather than blocks, which are like files in a simple file system
  - Objects are identified by a 64 bit Object ID (OID)
  - Objects are dynamically created and freed
  - Objects are variable length
- OSDS manage space allocation of objects
  - **OSD are not dump as conventional storage disks**

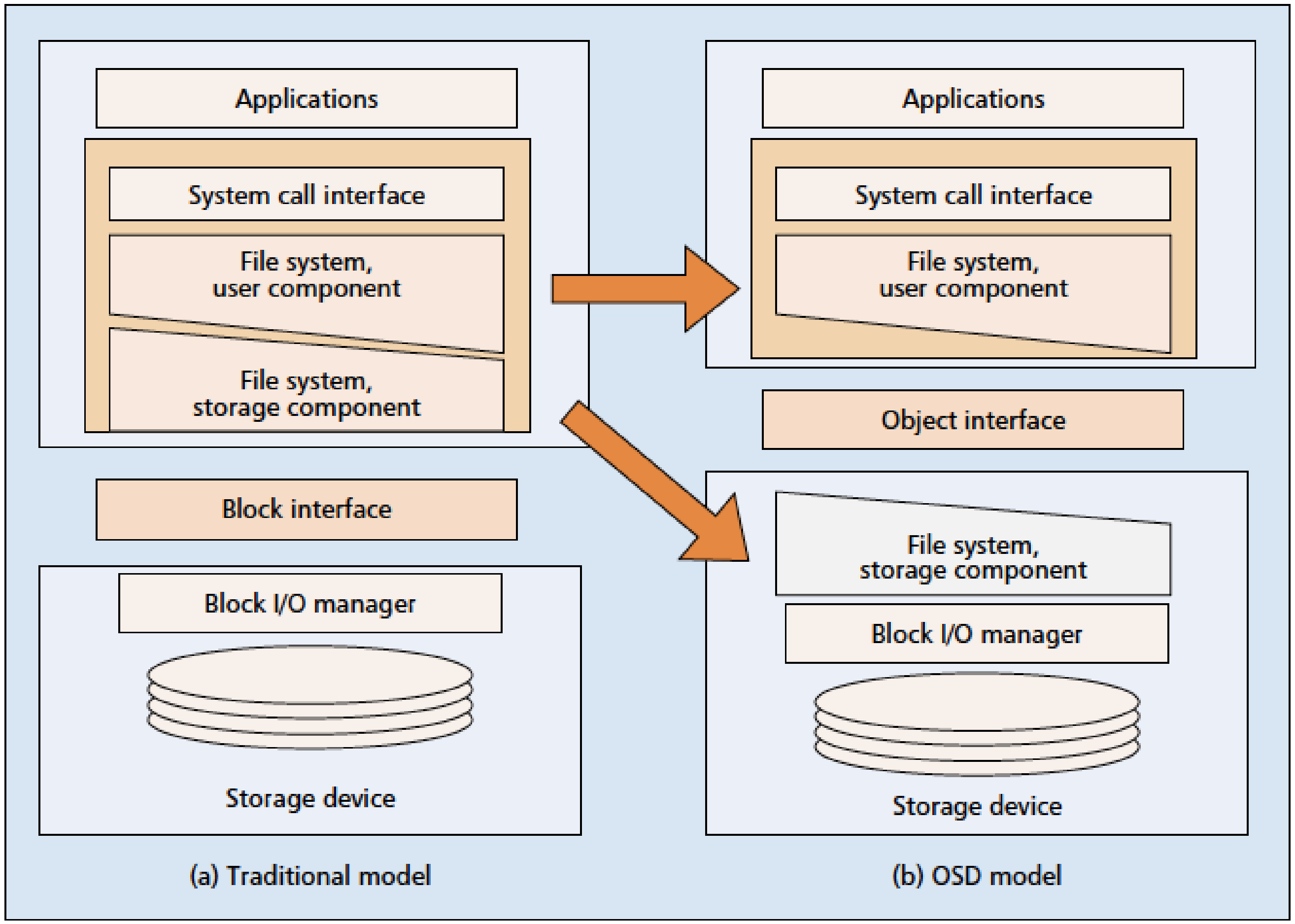
OSD vs. disk



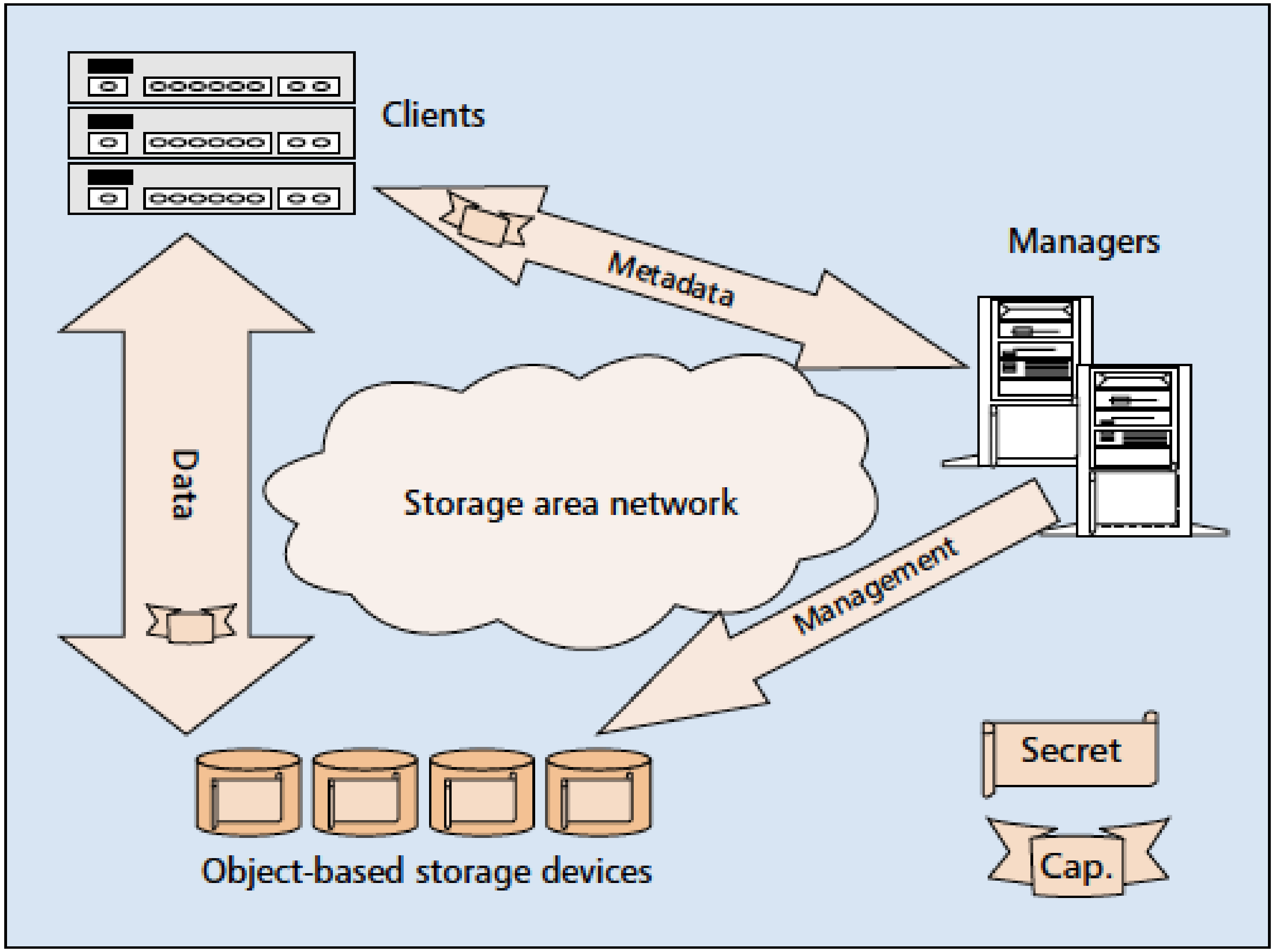
- Disk storage
  - Fixed array of blocks
- Disk operations
  - Read block, write block, format
- Disk security
  - Zoning and LUN Masking of entire disk
- Transport
  - FC SCSI and iSCSI

- OSD storage
  - Many objects of variable size
- OSD operations
  - Read object, write object, create object, list objects, etc.
- OSD security
  - On a per object basis using Capabilities
- Transport
  - FC SCSI, iSCSI, RPC

Offloading storage management from the file system



Object-based file system



Object-based file system

- Clients have direct access to storage to read and write file data securely
  - Contrast with SAN?
  - Contrast with NAS?
- File system includes multiple OSDs
  - Better scalability
- Multiple file systems share the same OSDs
  - Real storage pooling
- **File server is called the Metadata server (MDS)**

Design issues in distributed file systems

Design issues

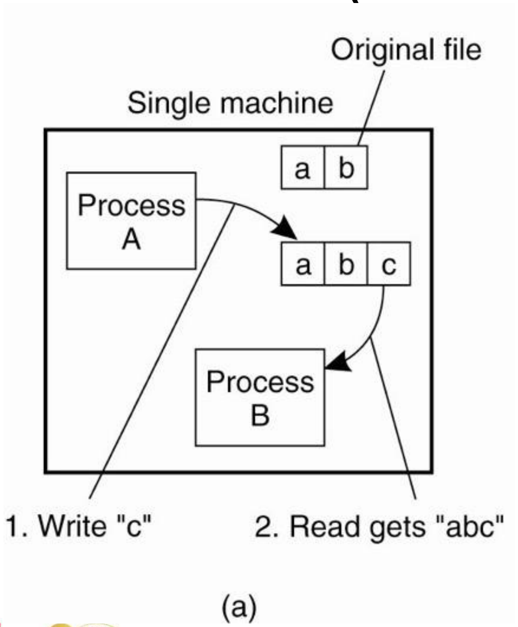
- Naming and name resolution
- Semantics of file sharing
- Caching
- Replication

Naming and name resolution

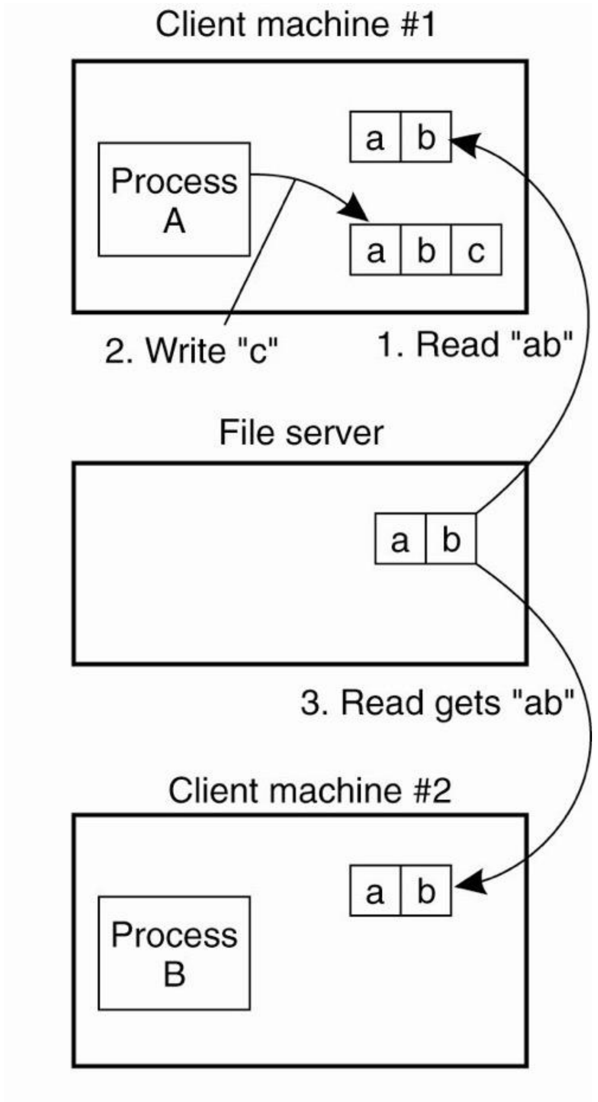
- A name space -- collection of names
- Name resolution -- mapping a name to an object
- 3 traditional ways
  - Concatenate the host name to the names of files stored on that host
  - Mount remote directories onto local directories
  - Provide a single global directory

File Sharing Semantics (1/2)

- Problem: When dealing with distributed file systems, we need to take into account the ordering of concurrent read/write operations, and expected semantics (=consistency).



(a)



(b)

File Sharing Semantics (2/2)

- Assume open; reads/writes; close
  - UNIX semantics: value read is the value stored by last write
    Writes to an open file are visible immediately to others that have this file opened at the same time. Easy to implement if one server and no cache.
  - Session semantics:
    Writes to an open file by a user is not visible immediately by other users that have files opened already.
    Once a file is closed, the changes made by it are visible by sessions started later.
  - Immutable-Shared-Files semantics:
    A sharable file cannot be modified.
    File names cannot be reused and its contents may not be altered.
    Simple to implement.
  - Transactions: All changes have all-or-nothing property.
    W1,R1,R2,W2 not allowed where P1 = W1;W2 and P2 = R1;R2

Caching

- Server caching: in main memory
    - cache management issue, how much to cache, replacement strategy
    - still slow due to network delay
    - Used in high-performance web-search engine servers
- Client caching in main memory
    - can be used by diskless workstation
    - faster to access from main memory than disk
    - compete with the virtual memory system for physical memory space
- Client-cache on a local disk
    - large files can be cached
    - the virtual memory management is simpler
    - a workstation can function even when it is disconnected from the network

Caching tradeoffs

- Reduces remote accesses => reduces network traffic and server load
  - Total network overhead is lower for big chunks of data (caching) than a series of responses to specific requests.
  - Disk access can be optimized better for large requests than random disk blocks
- Cache-consistency problem is the major drawback. If there are frequent writes, overhead due to the consistency problem is significant.

Replication

- File data is replicated to multiple storage servers
- Goals
  - Increase reliability
  - improve availability
  - balance the servers workload
- How to make replication transparent?
- How to keep the replicas consistent?
  - a replica is not updated due to its server failure
  - network partitioned

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!!!