CS5334.0251/0252, Spring 2010

CS5334, Advanced Internet Information Processing

Lecture 1: Introduction

(01-20-2010)

(Reading: Lecture Notes and listed references)

Lecture Outline

1. Data and data/file processing
   (1) Data and files
   (2) History of data processing
      a. File processing systems
      b. Problems with file processing systems
2. Database systems and database management systems
   (1) Database systems
   (2) Database management systems
   (3) Overview of database systems
3. Computer applications and services
   (1) Classes of computer applications & services
      a. Computation intensive applications
      b. Data intensive applications
      c. Data and computation intensive applications
      d. Other applications
   (2) Models of computer applications
      a. Centralized
      b. Client-server
      c. Process pool
      d. Peer process
      e. Proxy servers and caches
   (3) Variations of client-server architecture
      a. Mobile code
      b. Mobile agent
      c. Network computers
      d. Thin clients
   (4) Discussions
4. Information processing and Web applications

References:

- Chapter 1 of *Database System Concepts* (4nd ed.), A. Silberschatz, H. F. Korth, S. Sundarshan, McGraw-Hill, 2002.

- Chapter 2 of George Coulouris, Jean Dollimore, and Tim Kindberg, *Distributed Systems – Concepts and Design*, (3rd ed.). Addison-Wesley, 2001.

1. Data and data/file processing

(1) Data and files

(2) History of data processing

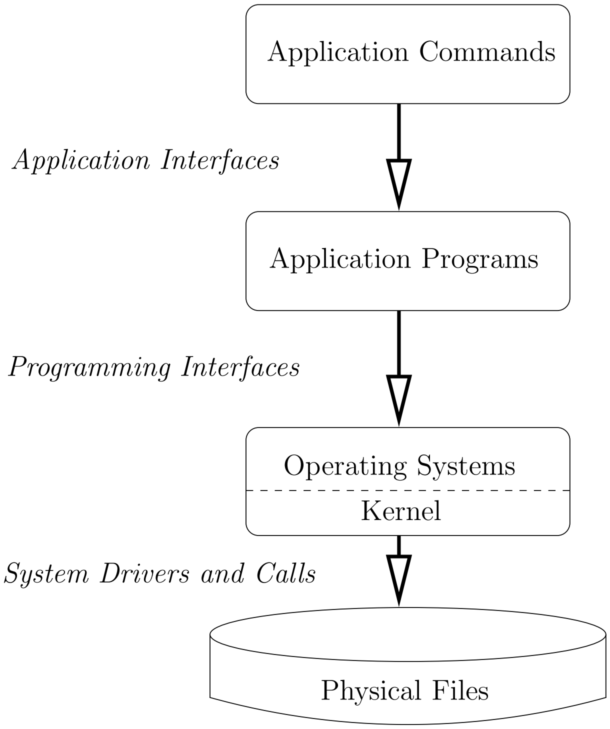


Figure 1: Illustration of File Processing Systems

a. File-processing systems (Fig.1)

(a) Features: a collection of files and application programs. These application programs have been developed over a period of time to meet ad hoc application demands.

(b) **Example 1**: Students information in a college or university: Name, ss#, gender, marriage status, birth date, home address, phone number, high school name. Main applications:

i. List all students;

ii. List all students with specific names;

iii. Lind all married students;

iv. Find all students who are younger than 18 years old.

(c) **Example 2**: A savings-bank enterprise that maintains information about customers and savings accounts. Various applications are possible:

i. A program to debit or credit an account;

ii. A program to add a new account;

iii. A program to find the balance of an account;

iv. A program to generate monthly statements.

Other possible applications include calculating and adding interest payments.

b. Problems with file-processing systems

(a) **Data redundancy and inconsistency:** Same piece of information may be duplicated in many different files. Example: name, ss#, phone number of a student may appear in both personal and academic files.

(b) **Difficulty in accessing data:**

(c) **Data isolation:** As data are scattered in various files, and files may be in different format, writing new application programs to retrieve the appropriate data and information could be quite challenging.

(d) **Integrity problems:** Data value must satisfy certain *consistency constraints*.

i. In the bank example, the balance of a savings account must be above certain value, such as $25.00.

ii. In the college example, the GPA of a student must be greater than or equal to certain value (such as 2.0). Otherwise the student may be put in probation or suspended.

(e) **Atomicity problems:** certain updating operations must be executed non-interrupted. Failures that occur in the middle of such updating operations may lead to data inconsistencies.

(f) **Concurrent-access anomalies:** access/updates by multiple users simultaneously may result in inconsistent date. Eg: the balance of a savings account may be invalid if the account is access by two transactions, one for depositing and another for withdrawing.

(g) **Security problem:** A department secretary should be only allowed to access information of students in her/his own dept.

2. Database systems and database management systems

(1) Database systems

a. General definition: A database system is a collection of interrelated files and a set of programs that allow multiple users to access and manipulate these files.

b. Emphasis: two components in a database system: the data (files) and control (programs).

(2) Database management systems

a. DBMS is the control portion of a DS. It consists of a set of programs that allow users and applications to access and manipulate the data in a DS. The DBMS provides a unified control over all data in a database system and presents a uniform view of data to the outside.

b. **Goals** of DBMS:

   (a) Minimize traffic between main memory and disk

   (b) Facilitate the access to the stored data. This includes easy to use system interfaces such as compilers and pre-compilers for embedded languages and interpreters.

c. **Responsibilities:**

   (a) Interaction with the disk storage. The raw data is stored on the disk using the file system that is usually supported by the OS. The DBMS translates various DML statements into low-level file system commands. Thus the DBMS is responsible for the actual storing, retrieving, and and updating of data in the database.

   (b) Integrity enforcement. The data in the database must satisfy certain types of consistency constraints. For example, the age of an employee must be greater than 18. Such constraints must be specified by the DBA. The DBMS is enforcing these constraints.

   (c) Security enforcement. Each access to the database must have appropriate privileges.

   (d) Back-up and recovery. Failures such disk crashes and power surges may cause loss of information. The database must periodically copied to back-up tapes. In case of failures, the DBMS must restore the database into a consistent state.

   (e) Concurrency control. Concurrent access to a single data record must be controlled to preserve consistency.

d. **Abstract views to users:** Hide the complex internal structure from users.

(3) Overview of database systems (Fig.2, DB)

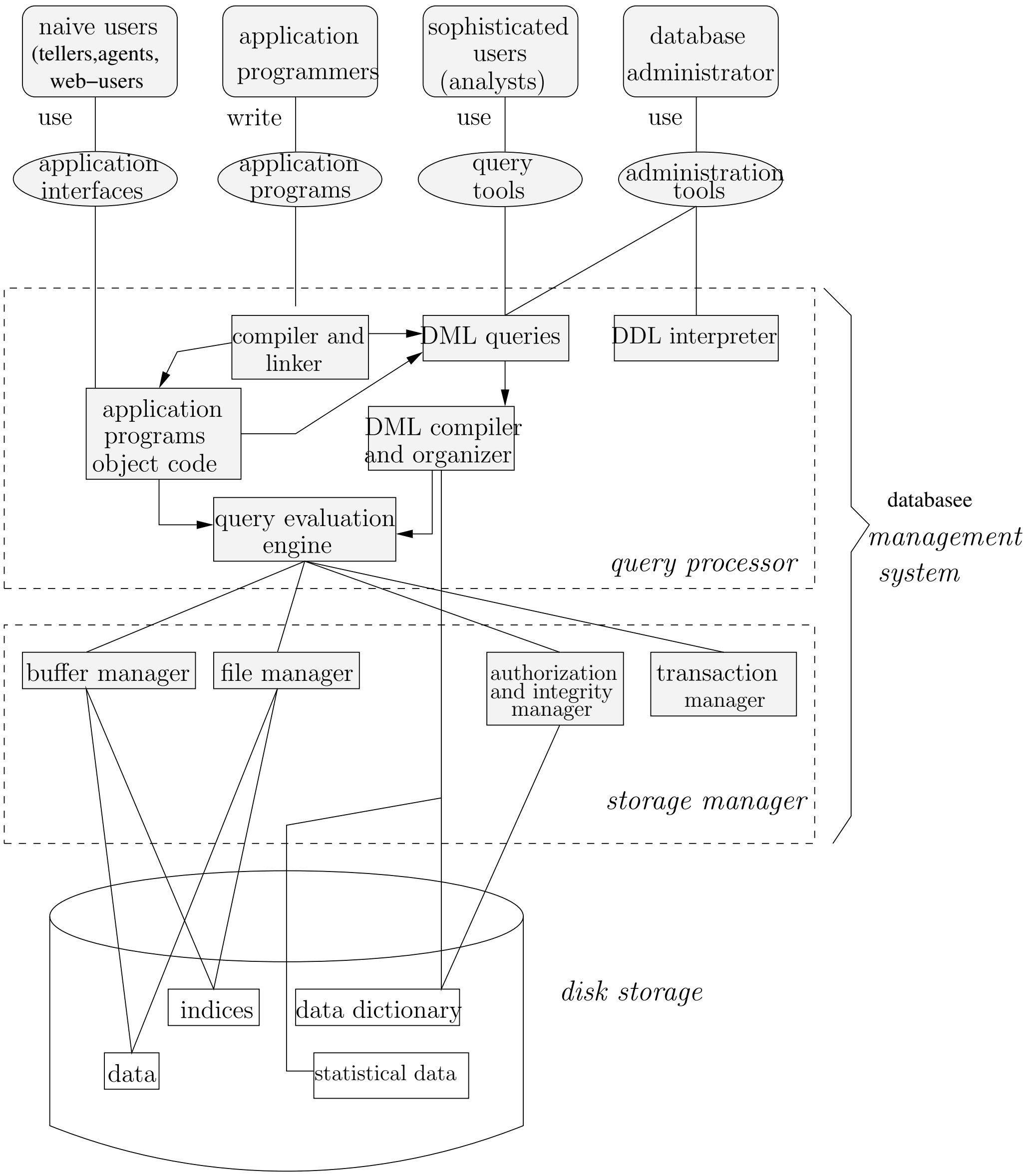


Figure 2: System structure (Fig.1.4, database textbook)

3. Computer applications and services

   (1) Classes of computer applications & services

      a. Computation intensive applications
         Examples:

         - ∗ Solving high-order differential equations
         - ∗ Solving numerical approximation problems

      b. Data intensive applications (often they are search related applications)

    - ∗ Searching for information of books containing a specific topic.

  c. Data and computation intensive applications

    - ∗ Whether forecast computation based on advanced mathematical models
    - ∗ Advanced web-based search with sophisticated rating criteria

  d. Other applications

    - ∗ Daytime service
    - ∗ Telnet service

(2) Models of computer applications

  a. Centralized model

    (a) A single such application is represented by a single process or a group of processes that cooperate to complete the desired task. All these processes are within the same computer system.

    (b) Advantages

      i. Applications are relatively easier to develop (no inter-machine communications are involved).

      ii. Typical for old days applications where all resources needed by applications were available locally within individual computer systems.

    (c) Problems: only applicable to applications whose required resources are all locally available.

  b. Client-server (Fig.3)

    (a) General ideas: control to individual resources is distributed among various processes across the Internet.

      i. Trend in OS design: reduce the size of OS kernel, and move part of the kernel up to higher levels (middleware) as system services. Processes are classified as **servers** and **clients**.

      ii. Each server process is in charge of a particular resource and client processes request the use of the resources through corresponding server processes.

      iii. Clients compete the use of resources

    (b) Advantages of client-server architecture

      i. The model is an natural extension of the centralized model to distributing applications over Internet. It is suitable for both centralized and distributed applications.

      ii. Separation of the client and server enables modular design strategies. Clear and compact interfaces provided by servers could facilitate client access and development.

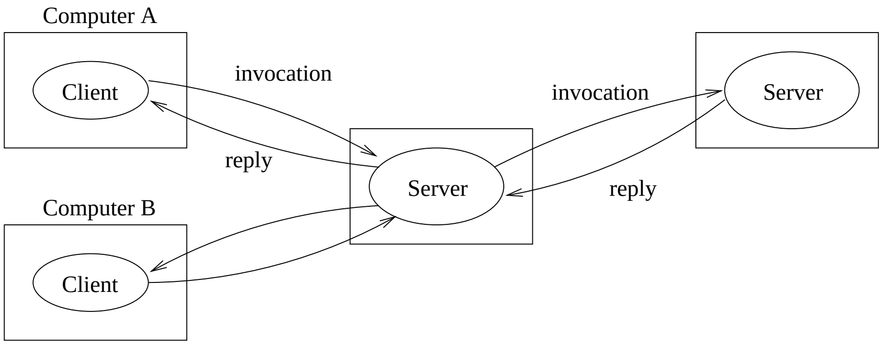


Figure 3: Clients invoke individual servers (Fig.2.2,DS)

iii. Due to its simplicity and intuitivety, it has become the most popular computation model.

(c) Problems with the client-server architecture

i. Control of individual resources is centralized in a single server. If the computer supporting the server fails, that resource becomes inaccessible. Compromise service reliability and availability.
Solution: duplicating services to multiple servers.

ii. Each server is a potential bottleneck, particularly when more clients demanding the same few resources are added. Subject to denial of service attacks.
Solution: duplicating services to multiple servers.

iii. Problems of duplicating servers to improve reliability and availability: Consistency maintenance; Resources demands.

c. Process pool (Fig.4)

(a) Basic architecture

i. A service may be implemented as several server processes in possibly different computers. These server processes may interact if necessary to provide services to individual client processes.

ii. The resources (objects) to be managed by the service could be partitioned among the servers. Alternatively, resources could be replicated to improve availability and performance

(b) Examples

i. Web services of large commercial organizations. Several Web servers running on different computers may be used (one for the main page, one for ads, another for video clips)

ii. SUN NIS (network information services) and other so called YP (yellow page) services that manage user passwords.

iii. DNS (domain name service).

(c) Advantages.

i. Could reduce probabilities of service bottleneck.

(d) Problems

i. The portion of a partitioned resource stored at an individual computer may become unavailable if that computer is down/ malfunctioning, or not reachable.

ii. Replicating frequently modified resources may actually hinder performance.

iii. More complex than the client-server model.

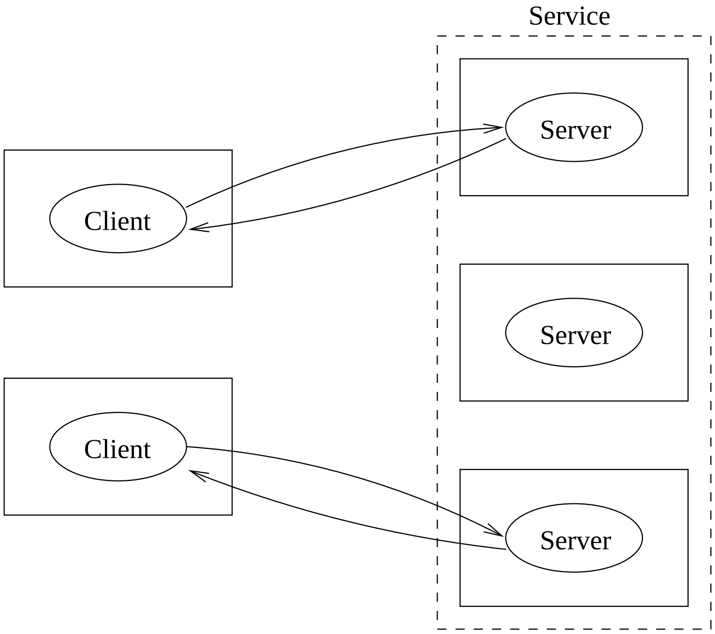


Figure 4: A service provided by a pool of servers (Fig.2.3, DS)

d. Peer process (Fig.5)

(a) Basic architecture

i. A group of processes playing similar roles interact with each other. They cooperate as *peers* to perform a distributed activity/computation. There is no clear distinction of clients and servers of these peer processes.

ii. The peer processes are coded in such a way that consistency of application-level resources (such as a replicated file) is maintained and actions on these resources are synchronized if necessary.

(b) Examples: *whiteboard* application: a group of programmers on several different computers view and modify an image. A peer process runs on each

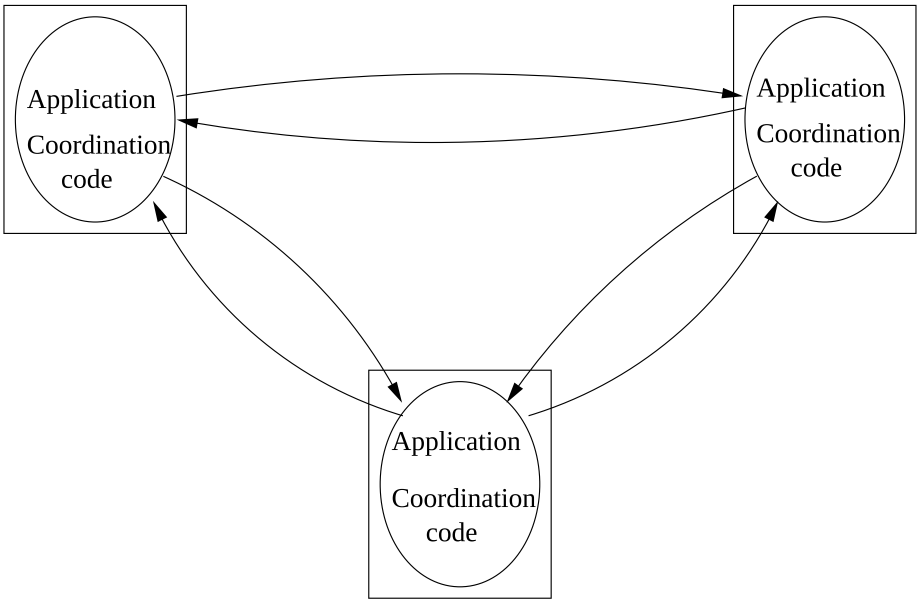


Figure 5: A distributed application based on peer processes (Fig.2.5, DS)

computer. Modification of the image by an individual programmer is communicated through middleware. Advantages: better response.

(c) Advantages: suitable and efficient for those special applications that fit the described architecture.

(d) Problems: limited in applicability

e. Proxy servers and caches (Fig.6)

(a) Architecture:

i. Cache: a store of recently referenced objects that are closer than the objects themselves.

ii. Web proxy servers: a server that keep a shared cache to be referenced by client sites.

iii. Proxy servers: allow access of shared cache through firewall.

(b) Characteristics: cached data is read only to client sites.

(c) Advantages and problems: only limited to the described special applications. Can reduce bandwidth demand.

(3) Variations of client-server architecture

a. Mobile code

(a) Definition: code that can be sent from one computer to another and run at the destination computer.

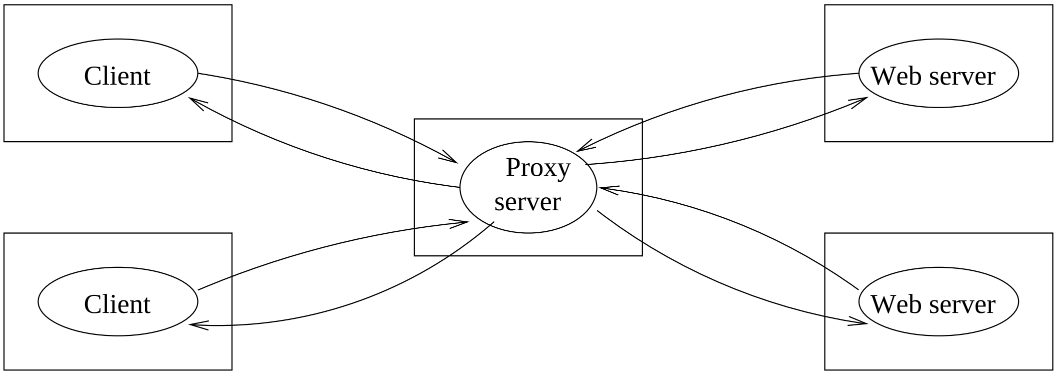
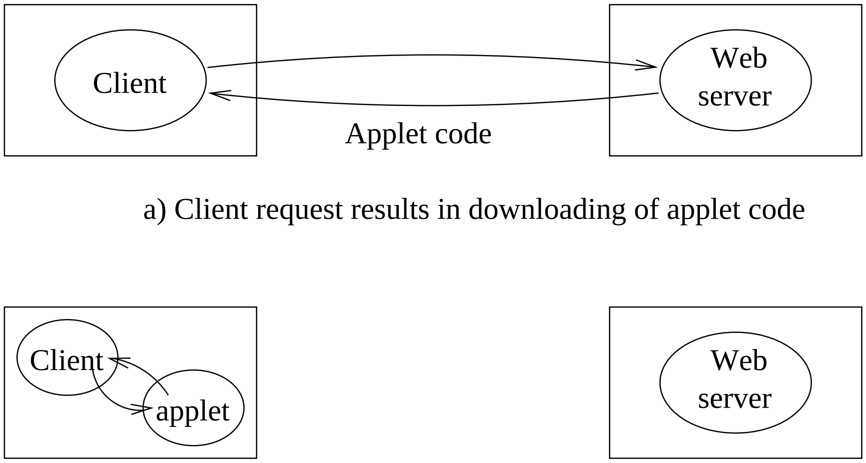


Figure 6: Web proxy server (Fig.2.5, DS)



a) Client request results in downloading of applet code

b) Client interacts wtih applet

Figure 7: Web applets (Fig.2.6,DS)

(b) Applet: a special mobile code that originally resides on a Web server. A client can select through a Web link the mobile code and download it from server and run it (Fig.6, DS).

(c) Advantage: running of the downloaded code locally by a client is much faster and smoother than running it in realtime on a server.

(d) Potential security problems

b. Mobile agent

(a) Definition: a mobile agent is a running program that can move (migrate) from one computer to another in a network while carrying task on behalf of some applications.

(b) Examples

- Software system maintenance

- · Information collection (user statistics in a DS)

(c) Potential security problems (similar as mobile agent)

c. Network computers

(a) Motivation: OS and application software require large amount of code. More importantly management and maintenance of the software require sophisticated skills.

(b) Definition: a network computer is one that does not have its own OS and any other application software. All of them have to be downloaded from remote servers. Applications are run locally while resources such as files are managed by remote servers. Users of NC only have to operate a remote controller similar to the one of a typical DVD player.

(c) Cache storage is used to reduce need of data transfer.

d. Thin clients

(a) Definition: a thin client is a software layer that supports a window-based user interface on a remote computer (Fig.8, DS)

(b) An X-Window software that runs on a window based PC. The software allows a user to interact with a remote X-window based UNIX computer on a local window based PC. The user can launch X-window based applications locally.

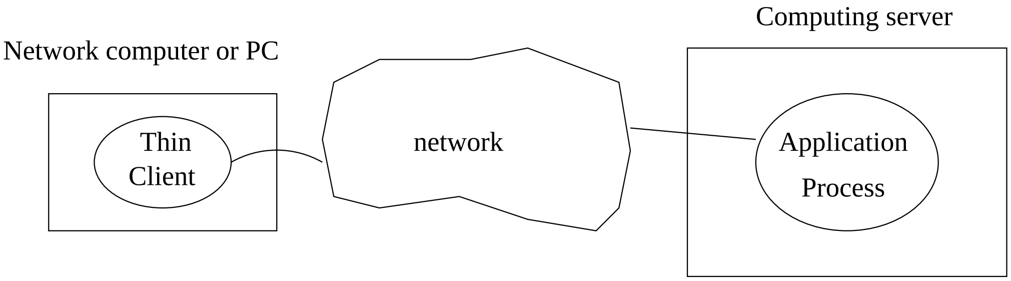


Figure 8: Thin clients and computing servers (Fig.2.7,DS)

(4) Discussions:

a. The selection of an architecture and the distribution of responsibilities between processes and between computers is an important issue in designing an application.

b. Often an application may adopt more than architectures to provide maximum flexibility and performance.

4. Information processing and Web applications

(1) From data processing/file processing to information processing

a. Information processing: a natural evolution of data/file processing

b. Characteristics of information processing

(a) Sophisticated processing algorithms

(b) Large amount of data/files

c. Database systems are essential

(2) Web applications

a. Characteristics of web applications

(a) Distributed processing

(b) The HTTP protocol provides an standard interface between web servers and clients

(c) Due to the standard interface development of web application clients (browsers) and servers is independent

(d) Can be both data/file and information processing

b. Impact of web applications

(a) Populating computers and their applications

(b) Advancing information processing