Software Architectures

Lecture 7: Client/Server Software Architecture

Professor:
Yushan (Michael) Sun
Fall 2020

Contents:

Concept on Client-server Software Architecture
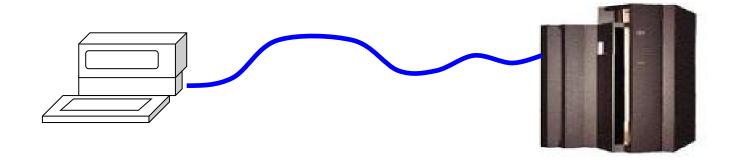
客户端-服务器架构的概念

1. Concept on Client-server Software Architecture
客户端-服务器架构的定义

定义: 客户端-服务器架构是一个多样化的、基于消息的、模块化的基础架构. 其目的是改善 Client/Server software architecture is a versatile, message-based and modular infrastructure that is intended to improve

- **可用性(Usability)** more efficient to use, easier to learn, more satisfying to use
- **灵活性(Flexibility)** designs that can adapt when external changes occur （电脑客户端，手机客户端）
- **互操作性(Interoperability)** the ability of diverse systems and organizations to work together (inter-operate)), and
- **可伸缩性(Scalability)** from 3,000 customers to 20,000 customers

1. Concept on Client-server Software Architecture

- **客户端(Client): A Requestor of Services, a piece of software**
- **服务器(Server): A Provider of Services, a piece of software**
- **交互方式：A client and a server communicate with each other (e.g. by SQL statements)**
- **硬件无关：Hardware (Physical separation) is not relevant**
- **平台无关：Platform (Wintel, Unix…) is not relevant**
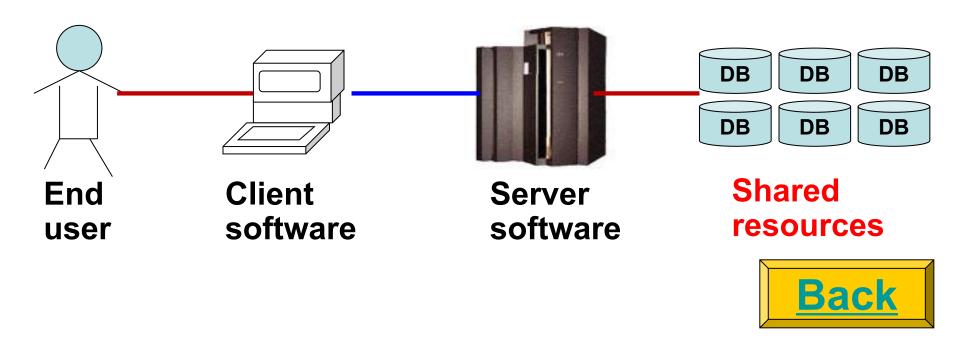
1. Concept on Client-server Software Architecture
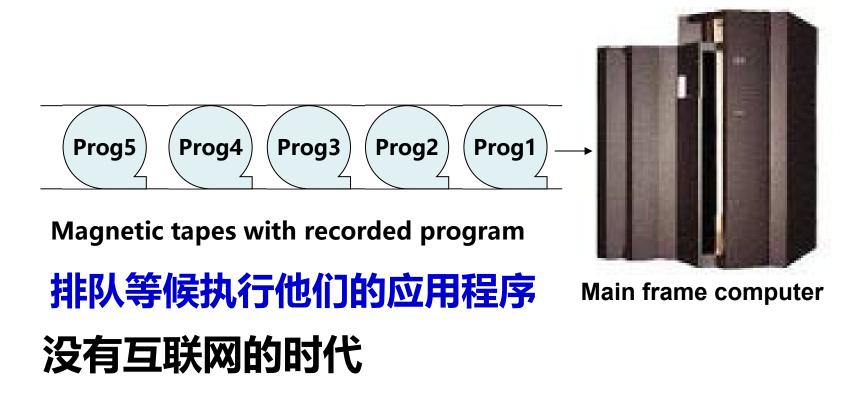
- **客户端-服务器架构的动机**

Motivation of client/server architecture:
- **To share resources among many users**
  - **client software interacts with end user**
  - **server software interacts with shared resource(s)**

End user	Client software	Server software	**Shared resources**

Back

Evolution of Client-server Architecture
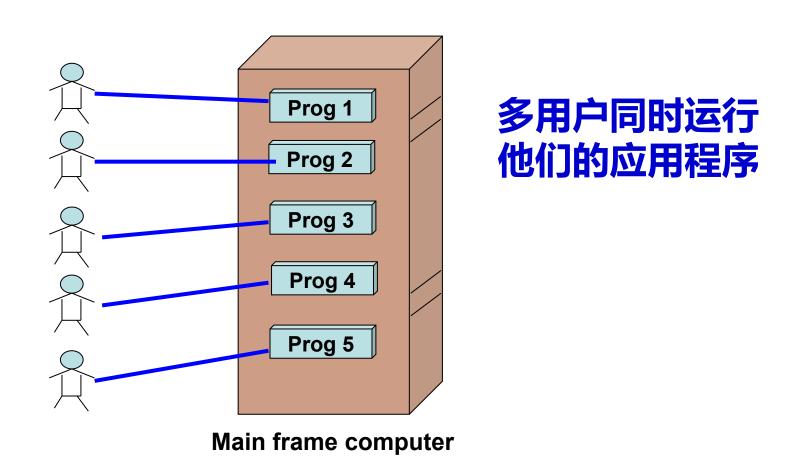
客户端-服务器架构的演化

2. Evolution of Client-server Architecture

一层客户端-服务器架构 (mainframe, -1970's):

- **From 1960's to 1970's, mainframe computers were used. Mainframe computer application can be regarded as one tire client/sever application. There are two ways for the mainframe to work.**

a) **批处理(Batch Processing): The batch processing queues up programs so that as soon as one completed, the next would start. To support a batch processing operation, some card punch writers would be used by programmers to write their programs "offline". Then the "program" in the punched card were submitted to the operations team, who would schedule them for running.**

2. Evolution of Client-server Architecture

Prog5 **Prog4** **Prog3** **Prog2** **Prog1** →

Magnetic tapes with recorded program

Main frame computer

排队等候执行他们的应用程序

没有互联网的时代

Batch Processing: programs are queued to wait for execution

2. Evolution of Client-server Architecture

b) **时间共享(Time-sharing). Time-sharing is the sharing of a computing resource among many users by means of multiprogramming and multi-tasking (concept: in 1960s, used widely in the 1970s)**

- **By allowing many users to interact concurrently with a single computer, time-sharing greatly lowered the cost of providing computing capability, made it possible for individuals and organizations to use a computer without owning one**

2. Evolution of Client-server Architecture

Prog 1

Prog 2

Prog 3

Prog 4

Prog 5

多用户同时运行他们的应用程序

Main frame computer
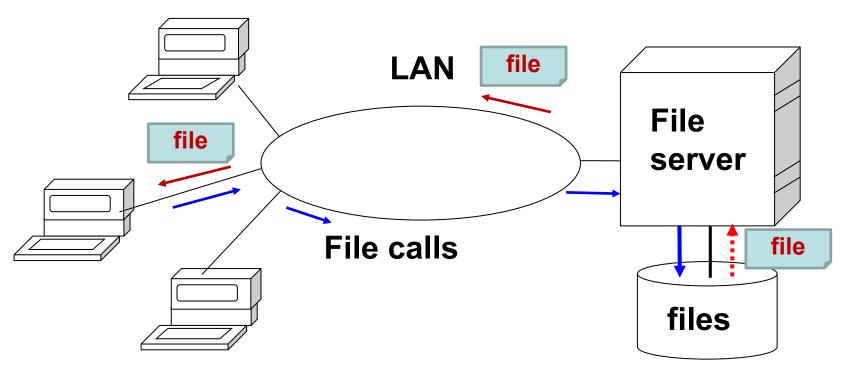
Time-sharing: allows a large number of users to interact concurrently with a single computer.

2. Evolution of Client-server Architecture

- **客户端与服务器的出现**
- **As time passed, computer prices were lower and lower, corporations separated mainframe-based applications into**
  - client and
  - server

  components

2. Evolution of Client-server Architecture

文件共享架构 (File Sharing Architecture)

a) PC requests file (records) from a file server
b) **Business logic all on client**, shared resources on server
c) Works when data volume are low

2. Evolution of Client-server Architecture

**Business logics
all on client**

LAN

file

file

**File
server**

File calls

file

files

Application clients

Sharing files across a network

2. Evolution of Client-server Architecture

具体步骤：

Application 1 fetches a file and then writes to the new line and then sends it back to the file server
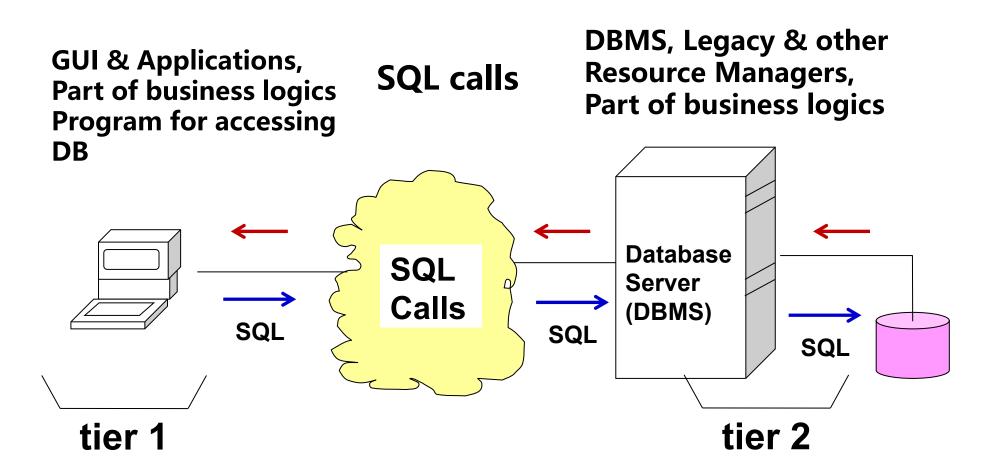
A file →

Sun, Mike	**356821**	**TV**	**888.56**
Lee, Mary	842674	radio	65.44
Juell, Paul	257215	Telephone	97.05
Nygard, Ken	573245	Microwave oven	56.88
Magel, Ken	834265	refrigerator	988
Hunt, Paul	332184	Bicycle	523
Eric, Vosika	257652	Watch	1005

2. Evolution of Client-server Architecture

Application 2 fetches a file and then writes to the new line and then sends it back to the file server

A file →

Liu, feng	221231	camera	506.00
Sun, Mike	356821	TV	888.56
Lee, Mary	842674	radio	65.44
Juell, Paul	257215	Telephone	97.05
Nygard, Ken	573245	Microwave oven	56.88
Magel, Ken	834265	refrigerator	988
Hunt, Paul	332184	Bicycle	523
Eric, Vosika	257652	Watch	1005

2. Evolution of Client-server Architecture

两层客户端-服务器架构

- **2-Tier Client/Server: Databases replaced File Server**
- **Word "Client/Server" traditionally associated with desktop PC connected over a network to a SQL-database server**
- **Server's database system (typically RDBMS) can answer a user's query (SQL)**
- **Advantage: Lowered network usage because sending queries and responses rather than files**

2. Evolution of Client-server Architecture

GUI & Applications, Part of business logics Program for accessing DB

SQL calls

DBMS, Legacy & other Resource Managers, Part of business logics

SQL

SQL
Calls

SQL

Database Server (DBMS)

SQL

tier 1

tier 2

2-Tier Client/Server: Databases replaced File Server

2. Evolution of Client-server Architecture

Contents of 2-Tier Client/Server

- **Tier 1**: GUI + Business Logic,
- **Tier 2**: Database + Shared Resources + Business Logic **(少量的程序)**

- 2-Tier is simple
  - Small departmental decision support
  - Simple Web publishing applications

2. Evolution of Client-server Architecture

- **应用程序被写在客户端或数据库服务器**
- **Application or business logic either resides on the client or on the database server in the form of stored procedures (存储过程)**
- **More complex applications**

  = fatter clients

  = more expensive PC's for each user
- **由PB和VB开发的数据库应用程序具有2层的客户端-服务器架构**
- **Database applications developed by Power Builder and VB have 2-tire client-server structure: front end + database**

2. Evolution of Client-server Architecture

- 两层客户端-服务器架构的优点:

- 与文件共享系统相比，因为两层客户端-服务器发送请求与回答而不是文件，从而降低了网络使用。

2. Evolution of Client-server Architecture

- **两层客户端-服务器架构的缺点：**
  - ➤ **局域网应用:** 它是单一服务器且以局域网为中心的，所以难以扩展至大型企业广域网或 Intranet；
  - ➤ **受限于供应商：** 程序的升级维护必须由供应商重新开发扩展；
  - ➤ **数据库开发商的锁定：** 因为你的GUI代码直接与DB服务器绑定。依赖于一个特定的开发商的技术使得你会很困难或者很高的费用使用别的开发商的产品，也就是说，更换其它产品困难。

2. Evolution of Client-server Architecture

➢ **程序配置困难**：当配置到远程的环境时候，两层的C/S结构管理起来比较困难。尤其是对于胖客户端的情况，就更难管理；

➢ **程序维护困难**：胖客户端导致程序维护困难：用户必须在客户端安装特定的客户端应用程序，而且当越来越多的企业的业务逻辑写在客户端应用程序中的时候，程序维护变得越来越困难。

➢ **软件升级困难**：需要每个客户端都要安装新的客户端应用程序
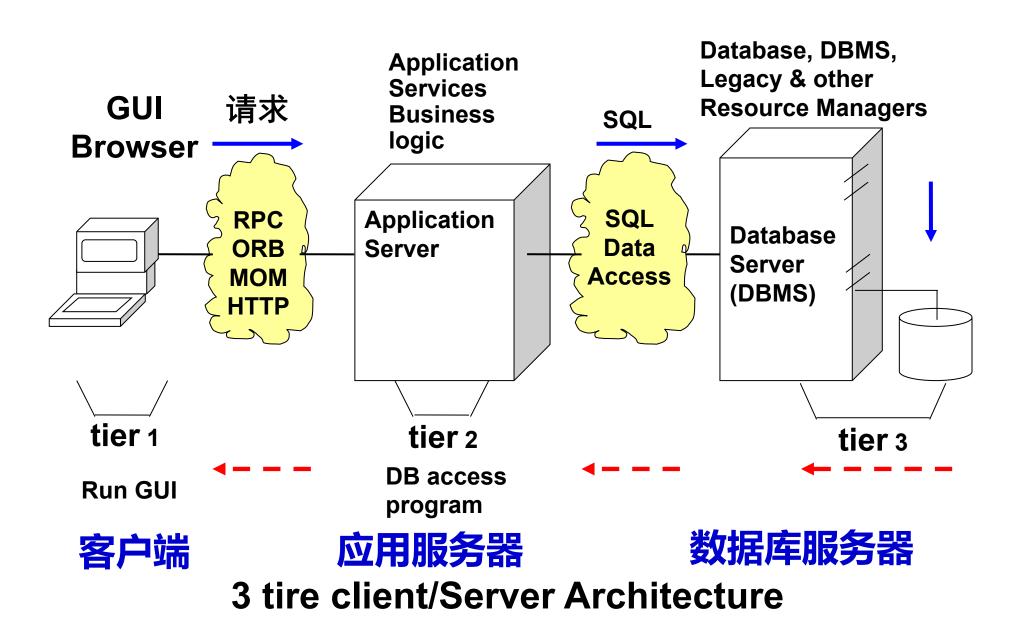
2. Evolution of Client-server Architecture

3层客户端-服务器架构
为了克服2层客户端-服务器架构的缺点，引进了3层客户端-服务器架构
To overcome the shortcomings of 2-tire client/server architecture, 3-tire client/server architecture is used as below
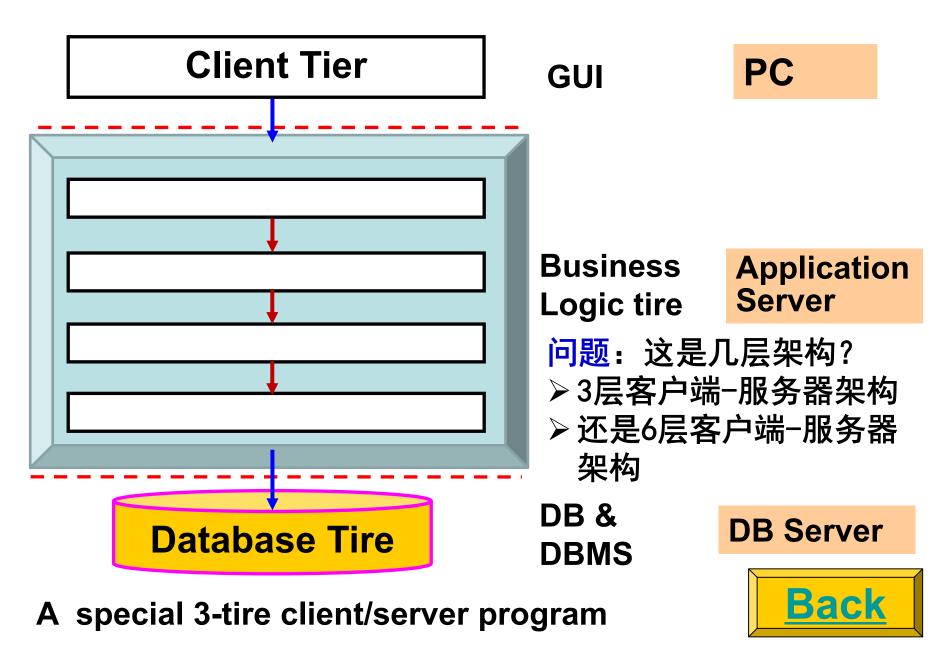
3-Tier Client/Server
- **Tier 1: GUI,**
- **Tier 2: Business Logic,**
- **Tier 3: Data**

- **Adds a middle tier between the client (UI) and the server (DBMS)**

2. Evolution of Client-server Architecture

GUI
Browser
请求

Application
Services
Business
logic

SQL

Database, DBMS,
Legacy & other
Resource Managers

RPC
ORB
MOM
HTTP

Application
Server

SQL
Data
Access

Database
Server
(DBMS)

tier 1

tier 2

tier 3

Run GUI

DB access
program

客户端

应用服务器

数据库服务器

3 tire client/Server Architecture

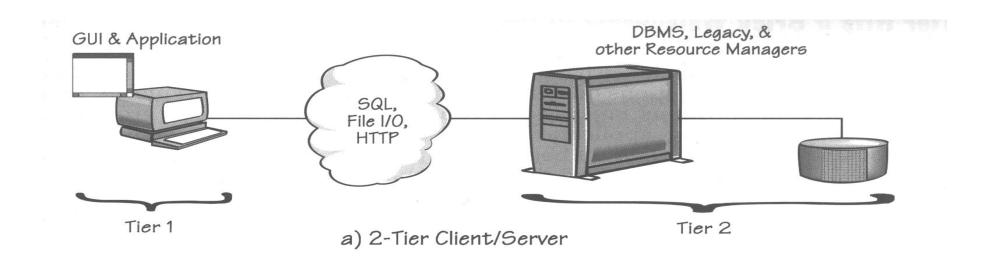## 2. Evolution of Client-server Architecture

- **3层客户端-服务器架构是最流行的架构**
- **3-tire client/server architecture is the most popular architecture**

- **所有的多层客户端-服务器架构(例如：JavaEE)均为由3层客户端-服务器架构扩充而来**
- **All other multiple client/server architecture are derived from the 3-tire client/server architecture**
- **The application layer in the 3-tire client/server design can be divided into several layers.**

2. Evolution of Client-server Architecture

Client Tier

GUI

PC

Business Logic tire

Application Server

问题：这是几层架构？
- ➤ 3层客户端-服务器架构
- ➤ 还是6层客户端-服务器架构

Database Tire

DB & DBMS

DB Server

Back

A special 3-tire client/server program
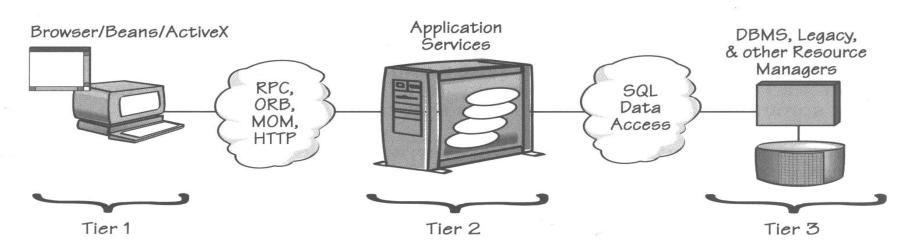
Comparison of 2- and 3-Tier Client-server Architecture

2层与3层客户端-服务器架构的比较

3. Comparison of 2- and 3-Tier client-server

GUI & Application

SQL, File I/O, HTTP

DBMS, Legacy, & other Resource Managers

Tier 1

Tier 2

a) 2-Tier Client/Server

Browser/Beans/ActiveX

RPC, ORB, MOM, HTTP

Application Services

SQL Data Access

DBMS, Legacy, & other Resource Managers

Tier 1

Tier 2

Tier 3

b) 3-Tier Client/Server

3. Comparison of 2- and 3-Tier client-server

Advantages (over 2 tire client server architecture):

1. **三层客户端-服务器架构更容易维护**。Application logic becomes separated from the GUI and the database; Changes in one tier should not affect the other tiers

2. **三层客户端-服务器架构可以有瘦客户端**。In mordern 3-tire client server architecture, all business logics are in the server, and the server is also responsible for generating web client, and so in the client machine, what is needed is windows operating system with browser. Using the browser, the web pages can be downloaded from the server to the client (**in this sense, the client tire is very thin**).

3. Comparison of 2- and 3-Tier client-server

3. **在三客户端-服务器架构中，各层开发语言可以是互相独立的**。

   3-tired client/server architecture has language independent property, e.g., Client could be written in Java, server could be written in C++.
   - Use Corba, in this case, to connect object in Java and object in C++

4. **三层客户端-服务器架构可以连接到多个数据库**。

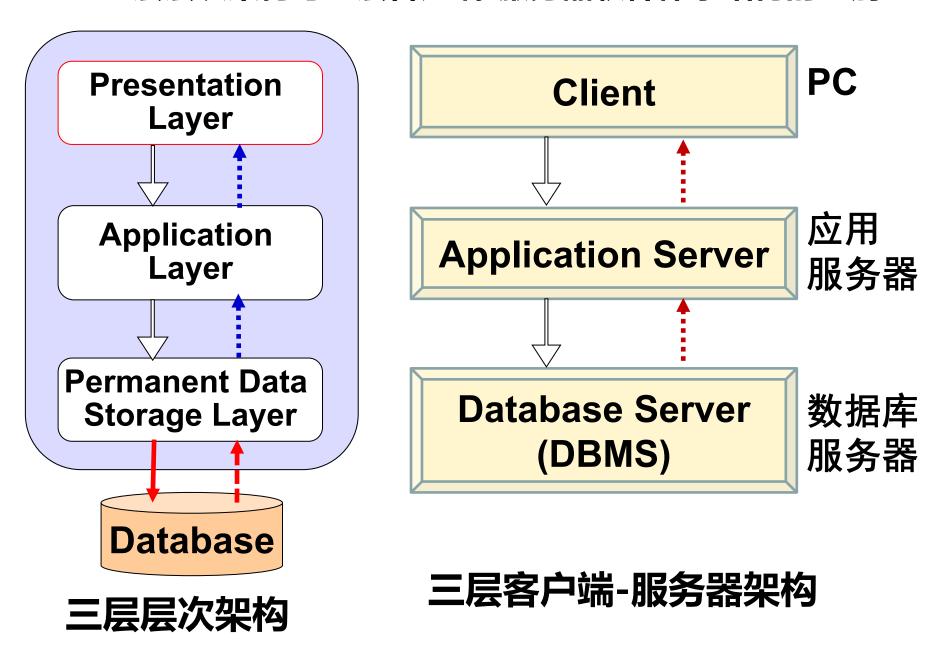   Can pool database connections

Back

Comparison of 3-layered Architecture and 3-tiered Client/server Architecture

3层层次架构与3层客户端-服务器架构的比较

4. 三层层次架构与三层客户端-服务器软件体系结构的区别

➢ **三层层次架构**（包括显示层，应用层与永久数据保持层）中的层指的是**逻辑意义上的层**。这些层，可以部署在同一台计算机上，当然也可以部署在不同的计算机上。

➢ **三层客户端-服务器架构**里面所涉及的层指的是**物理意义上的层**。

　• **客户端**运行在PC上；

　• **应用层**被部署与运行在应用服务器上；

　• **数据层**包含数据库与数据库管理系统软件，被部署与运行在数据库服务器上。

4. 三层层次架构与三层客户端-服务器软件体系结构的区别

Presentation Layer

Application Layer

Permanent Data Storage Layer

Database

三层层次架构

Client — PC

Application Server — 应用服务器

Database Server (DBMS) — 数据库服务器

三层客户端-服务器架构

三层客户端-
服务器架构

Browser/
Beans/
ActiveX

Application
Services
Business logic

Databse, DBMS,
Legacy & other
Resource Managers

三层客户端-
服务器架构

Browser/
Beans/
ActiveX

Presentation Layer

Business logic Layer

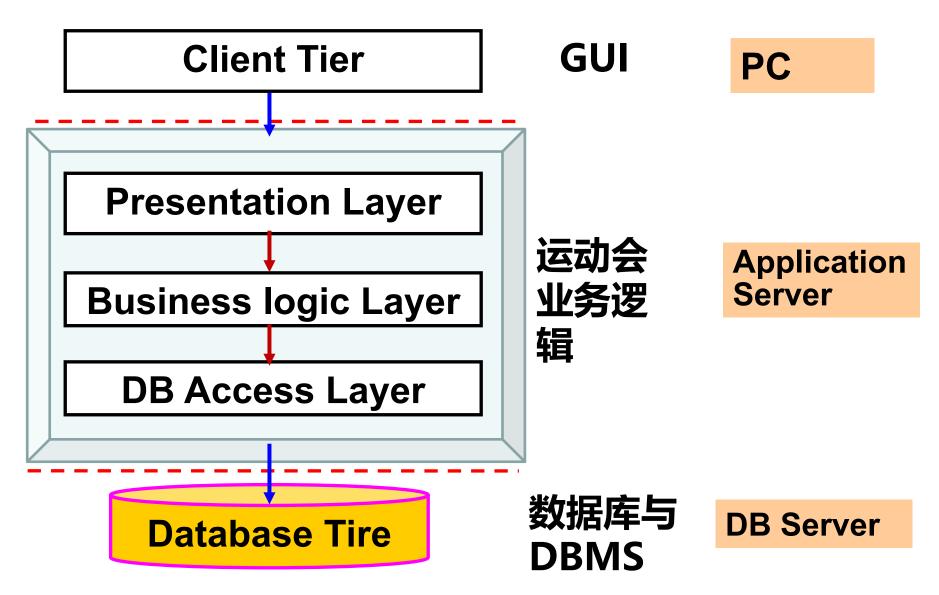DB Access Layer

Database, DBMS,
Legacy & other
Resource Managers

4. 三层架构与三层客户端-服务器软件体系结构的区别

- **例子：**

- 假如我们要利用三层客户端-服务器架构设计一个运行于Web上的简单的网上运动会管理系统。

- 则我们可能将软件系统设计为如下图所示的系统。

4. 三层架构与三层客户端-服务器软件体系结构的区别

| Client Tier | GUI | PC |

Presentation Layer

Business logic Layer

DB Access Layer

运动会业务逻辑

Application Server

Database Tire

数据库与DBMS

DB Server

An example design of a 3-tire client/server program

4. 三层架构与三层客户端-服务器软件体系结构的区别

- 注意到，在本设计中，对于三层的客户端-服务器架构的应用层部分，设计为一个通常意义下的三层的层次架构。

- 该架构包含显示层，用于生成用户图形界面，具体地说，图形界面生成代码被部署在服务层，但是生成的界面被显示在客户端PC上。

- 应用层包含所有的业务逻辑核心代码，永久数据保持层包含所有的数据库访问代码，例如JDBC代码。

- 实际上，层次架构里面的所谓的三个层都被部署在应用服务器上。因此，本设计可以被认为是应用了三层的客户端-服务器架构，其中在应用层中，包含了通常意义下的一个三层架构的程序。

4. 三层架构与三层客户端-服务器软件体系结构的区别

- **不要以为所有的程序都允许在互联网上**
- **在现实的软件开发中，除了运行在互联网上的程序外，还有很多程序不运行在互联网上，例如**
  - ➢ **Matlab软件，就是运行在桌面的软件；Matlab提供了强大的计算、仿真、绘图等功能因此，通常意义下的层次架构也具有广泛的用途**
  - ➢ **智能无人机控制软件**
  - ➢ **制造业软件**

JaveEE Architecture

5. JaveEE Architecture

Client Tire

运行在
客户机

Application Client Container

Application Client,Optional Java Beans

Web Client running in Web Browser, Optional Java Beans

Client computer

Web Tire

运行在
JavaEE
服务器

Web Container

Java Bean Components (optional)

Web Pages

Servlets

Java EE Server

Business tire

EJB Container

Java persistence Entities

Message Driven Beans

Session Beans

Back

运行在
数据库
服务器

EIS Tire

Database and Legacy system

Database Server