Multimedia Database Architecture

- Multimedia Architecture Requirements
  - ACID test
  - Multimedia Server Requirements
- Distributed Multimedia System
  - Super server concept
- Client-Server Systems
- P2P
- Media Streams
Multimedia Architecture Requirements

- Database architecture as a structure that facilitates the database to complete a transaction
- Four basic properties that a transaction should posses
  - Atomicity: All or nothing property. A transaction is an indivisible unit that is either performed or not
  - Consistency: A transaction must transform the database from one consistent state to another consistent state
  - Independence: Transactions execute independently of one another
  - Durability: The effects of a successfully committed transaction should be permanently recorded in the database

- ACID test of transaction reliability

ACID test

- For a single-user PC database where only one person is carrying out transactions at any one time the circumstances for the ACID test may be irrelevant
- Important for large number of users which access the database at the same time
  - A transaction can than only be achieved by locking the data rows involved to stop other users changing the data
  - Replicated database there may be more than one copy of the data that needs to be updated at the same time
Architecture of a multi-user database can become complex

- It is not clear which architecture would be the best option for a multimedia database
  - A transaction involving multimedia data will in general be expected to take longer
  - Locks will have to be maintained for longer periods

Formal database architecture

- Separate user view from the system view
- Three-layer architecture
- The external level provides the user's view of the database
  - It is a partial view
- The conceptual level is the community view of the database
  - Logical level as seen by the system administrator
  - In a relational database, relational conceptual level
- Internal level
  - The way the data is physically stored
  - In a relational database the internal level must not be relational
    - Records, pointers, etc..

- For multimedia objects, performance depends on the rate at which information can be transferred from storage memory for processing
- Block size affects the performance
  - Number of fetch operations
The architecture of the database system is influenced by the underlying computer and network system

- **Centralized database** system run on a single computer system that does not interact with other computer systems
- **Client-server system**, networking computers allow a division of work. Task relating to database structure are executed on server, presentation on the client computer
- **Distributed database systems** have been developed to handle geographically and administratively distributed data spread over multiple computer systems

**Figure 8.2** Multimedia databases – user, conceptual and physical storage views
Multimedia Server Requirements

- Often large scale applications
- Take into account:
  - User access behavior
  - Bandwidth
  - Storage requirements
    - (Complex multimedia formats)

Storage hierarchy

- Example, videos on demand:
  - High popular videos are stored in storage media with the highest bandwidth
**Characteristics**

- Minimal response time
- Reliability and availability
- Ability to sustain *guaranteed number of streams*
- Real-time delivery
- Exploit user access patterns

**Distributed Multimedia System**

- In a relational database that is distributed a table may be divided into a number of subrelations

- Horizontally - fragments consists of columns but only some rows
- Vertically - fragments consists of all rows but only some columns

- Partitioning of the data
Replicate fragments so that duplicates are stored on several sites

LOBs (video, music)) movements to a site, where they are likely to be requested (duplicates)
- Even daily basis!

Scalability
- Increasing number of users
- Size of data objects
- Amount of accessible data
  - Search, access, management
  - Non-uniform request distribution
Super server concept

- Distribute load among several servers
  - Problems arise when server selection is mainly based on systems defaults or on the user choice
  - This kind of static selection can cause uneven loads
- Dynamic server selection by alternatively mapping the servers in a local cluster
  - Saves local load problems

Super server concept

- Requests are directed to an appropriate server according to the location and the requested data, the current load of the servers, the location of the servers and the available network bandwidth
- User contacts a multimedia server as a normal server, and it makes the decision which is the most appropriate server
Client-Server Systems

- A special case of distributed systems
  - Certain sites are designated as clients and others as servers

- We will introduce
  - DataLinks as a specific art of SQL3
  - Development of intelligent middleware

DataLinks

- Store large unstructured data objects in a file system near a relational database
- Allows existing applications to incorporate multimedia with no changes to them
- Video and audio objects need to be streamed out to the client
  - Database servers do not have these capabilities
Intelligent Middleware

- Change information across systems developed by different vendors
  - Oracle, Ingres, DB2, MySQL
- Integration of information
- Three-tier systems were developed
  - Gateway to manage connections between the databases
- In large system there will be many servers
  - Data from local and external resources

Figure 8.7 Three-tier systems using mediators
Peer-to-Peer Networks

- Type of network in which each workstation has equivalent capabilities and responsibilities
- A peer-to-peer (P2P) application is different from traditional client-server model
  - Applications act both as client and server
- P2P networks are simpler
  - Low performance under heavy load

P2P application

- No central server
  - Napster (original), Freenet
- Discovering other peers
- Querying peers for content
- Sharing content with other peers
Heterogeneous Distributed DBMS

- Homogenous system all the sites use the same DBMS system
- Heterogeneous system different DBMS, different data models
Content Management

- Integration of a number of technologies

- Degree of semantics
  - Artifacts (date, location), content information (sentence, key shape, color histogram), domain concepts like ontologies
  - Decomposition of media into a database in terms of storage of metadata, building an indexing structure
  - should be an automatic process
Media Streams

- An important objective of multimedia systems design is to transfer data at a constant speed.
- Streaming is a technique for transferring data, so that it can be processed as a steady and continuous stream.
- By using streaming, the client browser can display the data before the entire file has been transmitted.

Definitions

- **Media stream**: the output of a sensor device such as a video, audio or motion sensor that produces a continuous or discrete signal.
- **Live multimedia**: the scenario where the multimedia information is captured in a real-life setting.
- **Continuous queries**: persistent queries that are issued once and then logically run continuously over live and unbounded streams.
Media Streams

- If the streaming client receives the data more quickly than required, it needs to be saved in a buffer
- However if the data does not arrive quickly enough, the presentation of the data will be not smooth

MM Networking Applications

Classes of MM applications:

1) Streaming stored audio and video
2) Streaming live audio and video
3) Real-time interactive audio and video
Fundamental characteristics:

- Typically **delay sensitive**
  - end-to-end delay
  - delay jitter

  **Jitter** is the variability of packet delays within the same packet stream

- But **loss tolerant**: infrequent losses cause minor glitches

---

**Streaming Stored Multimedia**

1. video recorded
2. video sent
3. video received, played out at client

**Streaming:** at this time, client playing out early part of video, while server still sending later part of video
Streaming Multimedia: Client Buffering

- Client-side buffering, playout delay compensate for network-added delay, delay jitter

Buffered video

Variable fill rate, \( x(t) \)

Constant drain rate, \( d \)

From network to decompression and playout

Streaming Multimedia: Client Buffering

- Client-side buffering, playout delay compensate for network-added delay, delay jitter
Quality-of-service Issue

(1) “Quality of service (QoS) is the collective effect of service performances which determine the degree of satisfaction of a user of the service”.

(ITU-T Recomm. E-800)

(2) A measure of the extent to which a user’s

“Quality of service represents the set of those quantitative and qualitative characteristics of a distributed multimedia system that are necessary to achieve the required functionality of an application.”

(Vogel, et al)
Degree to which some applications are Sensitive to Quality Parameters

<table>
<thead>
<tr>
<th>Application</th>
<th>Reliability</th>
<th>Delay</th>
<th>Jitter</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>File transfer</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Web access</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Remote login</td>
<td>High</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Audio on demand</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Video on demand</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Telephony</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Videoconferencing</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Bandwidth Requirements for Some Applications

<table>
<thead>
<tr>
<th>Application</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telephony</td>
<td>64 kbps</td>
</tr>
<tr>
<td>Telephony</td>
<td>16 - 32 kbps</td>
</tr>
<tr>
<td>Teleconferencing</td>
<td>48 - 64 kbps</td>
</tr>
<tr>
<td>2-channel audio</td>
<td>128 - 384 kbps</td>
</tr>
<tr>
<td>5-channel audio</td>
<td>320 kbps</td>
</tr>
<tr>
<td>Videoconferencing</td>
<td>64 - 153 kbps</td>
</tr>
<tr>
<td>High-Definition TV</td>
<td>17 Mbps</td>
</tr>
</tbody>
</table>
Characteristics of Multimedia Traffic

- Often consists of continuous streams of digitized video and audio data + bit-mapped images
- Video stream and audio stream are often sent separately
- Streams are broken up into discrete packets
- Packets are transmitted in sequence with uniform latency

Multimedia Network Requirements

**Throughput Requirements:**
- high transmission bandwidth
- large buffer capacity
- high-bandwidth channels for extended period of time
- error control
Tool for developing multimedia databases

Programming language:
- Indexing Methods
- Contend-based search
- Ontology
- Relational Database
- Interface: Web?

Solution:
- SQL+PHP?

---

Programming Language

- Java includes (JAI, JMF)
- JAI: Java Advanced Imaging API’s
  Functionality:
  - More than 100 image processing operations, most of which are native optimized for performance
  - Image Formats and Data Types:
    - Byte, UShort, Short, 32-bit int, floats/double, n-banded images
  - Image File I/O:
    - Supports BMP, GIF, FPX, JPEG, PNG, PNM, TIFF.
Java Media Framework

- The Java Media Framework API (JMF) enables audio, video and other time-based media to be added to applications and applets built on Java technology

- This optional package, which can capture, playback, stream, and transcode multiple media formats, extends the Java 2 Platform, Standard Edition (J2SE) for multimedia developers by providing a powerful toolkit to develop scalable, cross-platform technology

Database

- Relational Database
- How to Connect **Java** and a **Relational Database**?
  - (MySQL, Oracle)

- Java Platform, Enterprise Edition?
- WebObjects?
WebObjects

- WebObjects is an application server with tools, technologies, and capabilities to create Internet and intranet applications
  - Build applications that leverage the connectivity that the Internet or an intranet provide using the **client-server paradigm**
  - It has an object-oriented architecture that promotes quick development of reusable Web components
  - WebObjects is extremely scalable and supports high transaction volumes

Client-Server Application

- WebObjects allows you to develop two different types of Internet applications: Web applications and Java Client applications
  - Web applications are analogous to Common Gateway Interface (CGI) applications and consist of dynamically generated Web pages accessed through a Web browser
  - Java Client moves part of your application to the client-side computer and enlists Sun’s Java Foundation Classes (JFC) to give it the complete user interface found in a more traditional desktop application
Technology Overview: ASP.NET

- Successor of ASP
- Based on .NET framework
- Easier development and maintenance
- Enhanced performance
  - Compiled
  - Advanced caching
- Excellent dev tools: Visual Studio
- Enhanced productivity and security with ASP.NET 2.0

Technology Overview: ADO.NET

- Data access component
  - Manage connections
  - Data retrieval and manipulation
- Data structures independent of database
- Advanced data source controls with ADO.NET 2.0
Technology Comparison: Request Life Cycle

- **ASP.NET**
  - Page and controls objects created for every request (unless cached)
- **WebObjects**
  - Component object created once and put to sleep
  - Later requests awaken object and restore its content
  - When cache is full, pages used least recently removed

Technology Comparison: Frameworks & Platforms

- **ASP.NET**
  - Leverages .NET framework
    - Wide set of classes
    - Enhanced productivity as it targets Windows
    - Supports development in many languages
    - 2000 classes include in .NET 2.0
- **WebObjects**
  - Java-based: platform-independent
  - Tested and supported on Windows 2000, XP, Solaris, Unix-based systems
Technology Comparison:
Security

**ASP.NET**
- Forms and windows authentication
- Code Access Security
  - Code to be trusted to varying degrees
  - Developers can define what operations code can perform
- Role-based Security
  - ASP.NET 2.0 introduces profiles and role management

**WebObjects**
- Forms authentication and LDAP
  - Lightweight Directory Access Protocol

Technology Comparison:
Data Access Layer

**ADO.NET**
- Optimized for SQL Server
  - Disconnected data structures freshness (2.0)
- Different namespaces for databases
- Robust data structures
  - Typed DataSets
    - Can be extended to provide business logic
    - Tables, rows, columns, relationships accessible programmatically
    - But classes huge in size
    - Slower data transfers
- New controls violates good programming practices
Technology Comparison: Data Access Layer

- EO
  - Abstracts developer from data access layer
    - Database-agnostic approach
  - JDBC drivers may behave differently
  - Data retrieved as EOs
  - EOModeler creates EO_PK_Table to maintain primary keys

Example iTunes
Apple Store

- Disney
- Dell Computer
- BBC News
- ...
- Multimedia Architecture Requirements
  - ACID test
  - Multimedia Server Requirements
- Distributed Multimedia System
  - Super server concept
- Client-Server Systems
- P2P
- Media Streams