BCA-603
Mobile Computing
Unit III
Data Management Issues

A database is a collection of systematically stored records or information. Databases store data in a particular logical manner. A mobile device is not always connected to the server or network; neither does the device retrieve data from a server or a network for each computation. Rather, the device caches some specific data, which may be required for future computations, during the interval in which the device is connected to the server or network. Caching entails saving a copy of select data or a part of a database from a connected system with a large database. The cached data is hoarded in the mobile device database. Hoarding of the cached data in the database ensures that even when the device is not connected to the network, the data required from the database is available for computing.

What is Data Management?

Data management is the practice of collecting, processing, and using data securely and efficiently to achieve better business outcomes. It involves a broad range of tasks, policies, procedures, and practices to ensure that data is accessible, reliable, and usable for decision-making and operational purposes.

Key Principles of Data Management

1. **Data Collection and Integration**: This involves gathering data from various sources, both structured and unstructured, and integrating it into a unified system. This step is crucial for ensuring that data is available for analysis and decision-making.
2. **Data Storage**: Data can be stored in different types of repositories, such as databases, data warehouses, and data lakes. The choice of storage depends on the type of data and its intended use.
3. **Data Processing**: This includes cleaning, transforming, and organizing data to make it meaningful and useful. Techniques like data cleaning, aggregation, and enhancement are used to improve data quality.
4. **Data Security and Privacy**: Implementing measures to protect data from unauthorized access, breaches, and loss is essential. This includes data encryption, access control, and compliance with data privacy regulations.
5. **Data Governance**: This involves setting policies and procedures for data management, including data quality, access, usability, and security. Data governance ensures that data is managed consistently and responsibly across the organization.

6. **Data Lifecycle Management**: Managing the data lifecycle from creation to deletion, including data retention and disposal policies, is crucial for maintaining data quality and compliance.

Importance of Data Management

- **Informed Decision-Making**: Proper data management ensures that decision-makers have access to accurate and up-to-date information, enabling them to make informed choices.
- **Data Quality and Efficiency**: A well-managed data set reduces errors and improves efficiency, leading to better business outcomes.
- **Compliance and Security**: Effective data management helps organizations comply with regulations and protect sensitive data, reducing the risk of breaches and fines.
- **Competitive Advantage**: Organizations that manage their data effectively can gain insights into market trends and customer behavior, giving them a competitive edge.

Challenges in Data Management

- **Data Silos**: Data stored in isolated systems can hinder data integration and analysis.
- **Data Quality**: Ensuring data accuracy and consistency is a significant challenge.
- **Data Security**: Protecting data from unauthorized access and breaches is critical.
- **Compliance**: Adhering to data privacy regulations and standards requires robust data governance.

Conclusion

Data management is a critical practice for organizations looking to leverage their data for better decision-making and operational efficiency. By implementing effective data management strategies, organizations can ensure data quality, security, and compliance, ultimately driving better business outcomes

2. Topic: Hoarding techniques in mobile computing

In mobile computing, **hoarding** refers to prefetching and caching data on a mobile device before disconnection or in anticipation of poor network conditions. The goal is to improve performance, ensure availability, and reduce dependence on unreliable or expensive network connections. Here are some common hoarding techniques:

1. **Predictive Hoarding**

   - Uses **user behavior analysis** to determine which data will likely be needed.
   - Applies **machine learning** to track past usage patterns and predict future data requirements.
   - Example: A document editing app pre-downloads frequently accessed files before going offline.

2. **Context-Aware Hoarding**

- Takes into account factors like **location, time, and network conditions** to determine what data should be stored locally.
- Example: A **navigation app** downloads maps for an upcoming trip based on the user's calendar or GPS location.

3. **File Hoarding and Caching**

- Preloads frequently accessed **files, multimedia, or application data**.
- Example: **Streaming services** (Netflix, YouTube) allow users to download content for offline viewing.

4. **Database Hoarding**

- Syncs and caches structured data from **cloud databases** onto local storage for offline access.
- Example: **Google Drive, Dropbox, or Evernote** sync important files for offline use.

5. **Intelligent Prefetching**

- Dynamically fetches and updates data **in the background** before it is needed.
- Example: **Web browsers** prefetch webpages that a user is likely to visit next.

6. **Hoarding Based on Network Availability**

- Downloads and updates content when a **high-speed or free Wi-Fi connection** is available, avoiding cellular data use.
- Example: Smartphones automatically download app updates when connected to **Wi-Fi**.

7. **Application-Specific Hoarding**

- Tailored to specific applications, ensuring that **critical data is available** offline.
- Example: **Email apps** download recent messages and attachments so users can read them offline.

8. **Partial Hoarding**

- **Stores only the** most essential parts of a dataset (e.g., summaries, thumbnails, metadata) rather than the full data.
- Example: **News apps** download article headlines and short summaries instead of full articles.

Examples of Database Hoarding in Mobile Apps

Application Type	**Database Hoarding Use Case**
Email **Apps (Gmail, Outlook)**	Stores recent emails and attachments offline

Application Type	Database Hoarding Use Case
Cloud Storage (Google Drive, Dropbox)	Syncs frequently accessed files for offline use
E-commerce (Amazon, Flipkart)	Hoards shopping cart and product details
Travel Apps (Google Maps, Uber)	Caches maps, ride history, and itinerary details
Healthcare Apps	Stores patient records for offline access in remote areas

Data Caching in Mobile Computing

Data caching is the process of temporarily storing frequently accessed data on a mobile device to reduce latency, save bandwidth, and improve application performance. Cached data is typically stored in RAM, local storage, or specialized caching layers, allowing apps to retrieve information quickly without needing to fetch it repeatedly from a remote server.

Caching Strategies in Mobile Computing

Strategy	Description	Example
Write-Through Caching	Data is written to both the cache and database simultaneously.	A chat app updates messages in cache and database at the same time.
Write-Back Caching	Data is first written to the cache and later updated in the database.	A note-taking app saves drafts in cache before syncing to cloud storage.
Cache Invalidation	Ensures stale data is removed when updates occur.	A weather app refreshes cached forecasts when new data is available.
Least Recently Used (LRU) Eviction	Removes the least accessed data when cache space is full.	A music streaming app removes least played songs from cache.
Time-to-Live (TTL) Expiration	Data expires after a predefined time period.	A stock market app refreshes prices every 10 seconds.

Benefits of Data Caching

✅ **Reduces Latency** → Faster app performance and responsiveness.
✅ **Saves Bandwidth** → Fewer network requests reduce data usage.
✅ **Enhances Offline Access** → Cached data allows limited functionality without an internet

connection.
✅ **Improves Scalability** → Reduces server load and improves app efficiency.

Data Prefetching in Mobile Computing

Data prefetching is a technique used in mobile computing to **proactively fetch and store data** before it is needed. This reduces latency, improves user experience, and ensures smooth app performance, especially in environments with intermittent or slow network connections.

Prefetching Strategies

Strategy	Description	Example
User Behavior-Based Prefetching	Analyzes user habits and fetches data accordingly.	A reading app loads the next chapter of a book before the user reaches the end.
Application-Specific Prefetching	Prefetching tailored to app functionality.	A music app loads upcoming songs in a playlist.
Hardware-Aware Prefetching	Adjusts prefetching based on device battery and CPU load.	A cloud storage app delays large file syncs when the battery is low.
Cache-Assisted Prefetching	Works with caching to avoid redundant requests.	A social media app prefetches images while scrolling.

Benefits of Data Prefetching

✅ **Reduces Load Times** → Data is ready before the user needs it.
✅ **Improves Offline Access** → Users can access content even without an active connection.
✅ **Optimizes Network Usage** → Prefetching over Wi-Fi reduces mobile data costs.
✅ **Enhances User Experience** → Seamless browsing and faster app responsiveness.

3 Topic: Data Replication in Mobile Computing

Data replication is the process of copying and maintaining **multiple synchronized copies** of data across different devices, servers, or locations. In mobile computing, it ensures **data consistency, availability, and fault tolerance**, especially in environments with intermittent or unreliable network connections.

Types of Data Replication

1. Full Replication

- Entire database is copied to multiple locations.
- **Example:** Cloud storage services like **Google Drive** replicate user files across multiple servers for backup and availability.

2. Partial Replication

- Only a subset of the data is replicated, based on user preferences or app requirements.
- **Example:** A **news app** stores recent articles locally, while older ones remain in the cloud.

3. Synchronous Replication

- Changes are instantly copied to all replicas.
- Ensures strong consistency but can be **slow** due to real-time synchronization.
- **Example: Banking applications** where transactions must be updated in all locations simultaneously.

4. Asynchronous Replication

- Changes are updated in batches or at scheduled intervals.
- Improves performance but may cause temporary inconsistencies.
- **Example: Email apps** sync new emails only when connected to Wi-Fi.

5. Peer-to-Peer Replication

- Multiple devices or servers share data directly without a central authority.
- **Example: Decentralized apps (DApps)** and some **blockchain networks** use peer-to-peer replication.

6. Master-Slave Replication

- One primary database (master) handles all updates, while secondary databases (slaves) receive copies.
- **Example: Cloud-based applications** where a central server manages updates while mobile devices sync periodically.

Key Benefits of Data Replication

✅ **Improved Data Availability** → Ensures access even if the main server is down.
✅ **Faster Data Access** → Users get data from the nearest replica, reducing latency.
✅ **Offline Access** → Apps function without an internet connection and sync later.
✅ **Load Balancing** → Spreads traffic across multiple servers, preventing overload.
✅ **Disaster Recovery** → Protects data from loss due to hardware failures.

Topic: Adaptive clustering for mobile wireless networks

Adaptive Clustering in Mobile Wireless Networks

Adaptive clustering is a dynamic clustering technique used in **mobile wireless networks** to enhance network efficiency, scalability, and stability. It groups mobile nodes into clusters based on real-time network conditions and adapts to changes like mobility, traffic load, and energy constraints.

Key Features of Adaptive Clustering

1. **Dynamic Cluster Formation**
   - Nodes are grouped into clusters **dynamically** based on proximity, signal strength, or communication needs.
   - **Example:** A **vehicular ad hoc network (VANET)** where moving vehicles form and dissolve clusters based on traffic flow.
2. **Cluster Head Selection**
   - A **Cluster Head (CH)** is selected to manage communication within the cluster.
   - CH selection is adaptive, considering factors like **battery life, mobility, and processing power**.
3. **Self-Healing & Reconfiguration**
   - If a Cluster Head moves away or fails, another node takes over **without disrupting communication**.
   - Ensures continuous connectivity in highly dynamic environments.
4. **Load Balancing**
   - Workload is **distributed among clusters** to prevent congestion.
   - Example: In a **sensor network**, energy-efficient nodes are prioritized to extend network lifespan.
5. **Energy Efficiency**
   - Reduces energy consumption by minimizing **direct communication with the base station**.
   - Ideal for **battery-powered mobile devices** like IoT sensors and drones.

Adaptive Clustering Algorithms

Algorithm	Description	Use Case
Lowest-ID Algorithm	Selects the node with the lowest ID as the Cluster Head.	Simple but may not balance energy use efficiently.
Highest-Degree Algorithm	The node with the most neighbors becomes the Cluster Head.	Works well in dense networks but can overload CHs.

Algorithm	Description	Use Case
Weighted Clustering Algorithm (WCA)	Uses mobility, battery life, and connectivity to select CHs.	Used in MANETs, VANETs, and IoT networks.
LEACH (Low-Energy Adaptive Clustering Hierarchy)	Rotates CHs periodically to balance energy use.	Ideal for wireless sensor networks (WSNs).

Applications of Adaptive Clustering

✅ **Mobile Ad Hoc Networks (MANETs)** → efficient routing in decentralized networks.
✅ **Vehicular Ad Hoc Networks (VANETs)** → Cluster-based communication for traffic safety.
✅ **Wireless Sensor Networks (WSNs)** → Energy-efficient data aggregation in IoT applications.
✅ **5G & IoT Networks** → reducing network congestion and improving resource management.
✅ **Disaster Recovery & Military Networks** → Reliable communication in unpredictable environments.

Adaptive Clustering for Mobile Wireless Networks and File Systems

Adaptive clustering is a technique used in **mobile wireless networks** to organize mobile nodes into dynamically adjusting clusters for efficient communication. When applied to **file systems**, it helps in **data distribution, replication, and access optimization** in mobile environments.

---

1. Adaptive Clustering in Mobile Wireless Networks

Key Concepts

✅ **Cluster Formation:** Mobile nodes form clusters based on parameters like **mobility, signal strength, and energy levels**.
✅ **Cluster Head Selection:** A **Cluster Head (CH)** is dynamically chosen to manage communication within the cluster.
✅ **Load Balancing:** Reduces network congestion by distributing tasks among nodes.
✅ **Self-Healing & Reconfiguration:** Clusters adapt when nodes move or leave the network.

Benefits

✓ **Enhances Network Scalability** → Reduces routing overhead and improves performance.
✓ **Energy Efficiency** → Saves battery power by limiting long-distance communication.
✓ **Minimizes Interference** → Organizes communication to prevent collisions.

Example Use Cases

📌 **Vehicular Ad Hoc Networks (VANETs)** → Adaptive clustering helps organize moving vehicles for traffic communication.
📌 **Wireless Sensor Networks (WSNs)** → IoT sensors use clustering for efficient data transmission.
📌 **5G Networks** → Cluster-based communication improves resource management and bandwidth efficiency.

---

2. Adaptive Clustering in Mobile File Systems

Key Concepts

✅ **Data Replication & Caching:** Frequently accessed files are replicated across clusters to improve availability.
✅ **Distributed File Storage:** Clusters store different file segments, reducing redundancy and improving retrieval speeds.
✅ **Dynamic Cluster Reconfiguration:** When a node moves, its file system adapts by redistributing stored data.
✅ **Energy-Aware Data Management:** Data access operations are scheduled to reduce energy consumption on mobile devices.

Benefits

✓ **Improved File Access Speed** → Cached and replicated files ensure faster retrieval.
✓ **Fault Tolerance** → If a node fails, another node in the cluster has a backup.
✓ **Optimized Bandwidth Usage** → Reduces unnecessary file transfers over the network.

Example Use Cases

📌 **Cloud-Assisted Mobile File Systems** → Cloud storage apps (Google Drive, Dropbox) optimize file access using adaptive clustering.
📌 **IoT Edge Computing** → Sensor networks distribute data storage across clustered devices.
📌 **Mobile Peer-to-Peer (P2P) File Sharing** → Clustering improves data exchange efficiency in decentralized networks.