Unit-III

What is mobile database?

Mobile databases are separate from the main database and can easily be transported to various places. Even though they are not connected to the main database, they can still communicate with the database to share and exchange data.

The mobile database includes the following components −

- The main system database that stores all the data and is linked to the mobile database.
- The mobile database that allows users to view information even while on the move. It shares information with the main database.
- The device that uses the mobile database to access data. This device can be a mobile phone, laptop etc.
- A communication link that allows the transfer of data between the mobile database and the main database.

Advantages of Mobile Databases

Some advantages of mobile databases are −

- The data in a database can be accessed from anywhere using a mobile database. It provides wireless database access.
- The database systems are synchronized using mobile databases and multiple users can access the data with seamless delivery process.
- Mobile databases require very little support and maintenance.
- The mobile database can be synchronized with multiple devices such as mobiles, computer devices, laptops etc.

Disadvantages of Mobile Databases

Some disadvantages of mobile databases are −

- The mobile data is less secure than data that is stored in a conventional stationary database. This presents a security hazard.
- The mobile unit that houses a mobile database may frequently lose power because of limited battery. This should not lead to loss of data in database.

Data Management issues in Mobile database

Data management technology that can support easy data access from and to mobile devices is among the main concerns in mobile information systems. Mobile computing may be considered a variation of distributed computing. The two scenarios in which mobile databases is distributed are: Among the wired components, the entire database is distributed, possibly with full or partial replication. A base station or fixed host manages its own database with a DBMS like functionality, with additional functionality for locating mobile units and additional query and transaction management features to meet the requirements of mobile environments.

Among the wired and wireless components, the database is distributed. Among the base stations or fixed hosts and mobile units, the data management responsibility is shared. Here are some of the issues which arises in **data management** of the mobile databases:

1. Mobile database design –

The two fundamental design issues are fragmentation, the separation of the database into partitions called fragments, and distribution, the optimum distribution of fragments. The research in this area mostly involve mathematical programming in order to minimize the combined cost of storing the database, processing transactions against it, and message communication among site.

2. Security –

The data which is left at the fixed location is more secure as compared to mobile data. That is mobile data is less secure. Data are also becoming more volatile and techniques must be able to compensate for its loss. The most important thing needed in this environment is the authorizing access to critical data and proper techniques.

3. Data distribution and replication –

Uneven distribution of data among the mobile units and the base stations take place here. Higher data availability and low cost of remote access is there in data distribution and replication. The problem of Cache management is compounded by the consistency constraints. The most updated data and frequently accessed data is provided by the Caches to the mobile units. It process their own transactions. There is most efficient access of data and higher security is available.

4. Replication issues –

There is increase of costs for updates and signalling due to increase in number of replicas. Mobile hosts can move anywhere and anytime.

5. Division of labour –

There is a certain change in the division of labour in query processing because of certain characteristics of the mobile environment. There are some of the cases in which the client must function independently of the server.

6. Transaction models –

In mobile environment, the issues of correctness of transactions and fault tolerance are aggravated. All transactions must satisfy the ACID properties, these are atomic, consistency, isolation, and durability. Depending upon the movement of the mobile unit, possibly on multiple data sets and through several base station, a mobile transaction is executed sequentially. When the mobile computers are disconnected, ACID properties gets hard to enforce. Because of the disconnection in mobile units, there is expectation that a mobile transaction will be lived long.

7-Recovery and fault tolerance –Fault tolerance is the ability of a system to perform its function correctly even in the presence of internal faults. Faults can be classified in two types: transient and permanent. Without any apparent intervention, a transient fault will be eventually disappeared but a permanent fault will remain unless it is removed by some external agency.

The mobile database environment must deal with site, transaction, media, and communication failures. Due to limited battery power there is a site failure at MU. If a voluntary shutdown occurs in MU, then it should not be treated as a failure. Whenever MU crosses the cells, most frequently there will be a transaction failures during handoff. Due to failure of MU, there is a big cause of network partitioning and affection of the routing algorithms. The characterization of mobile computing is done by:

- Limiting resource availability

- Frequent disconnection
- High mobility
- Low bandwidth

8. Location based service –

One of the most challenging tasks which must be undertaken is determining the location of mobile users, which must be undertaken in order to enable a location based service. A cache information becomes sale when clients move location dependent. Eviction techniques are important in this case. Issues that arises in location and services are:

- User Privacy
- Diverse mobile mapping standards
- Market capability
- Interoperability

Updation of the location dependent queries and then applying spatial queries to refresh the cache causes a problem.

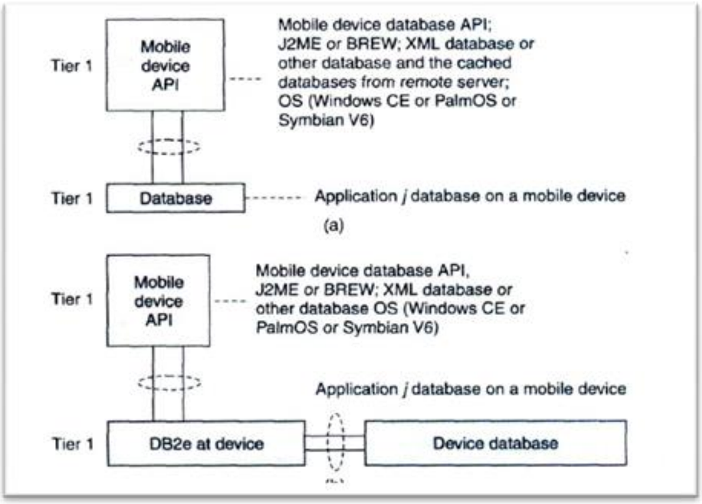
9. Query processing –

Because of the mobility and rapid resource changes of mobile units, Query optimization becomes the most complicated. That is query processing is affected when mobility is considered. There is a need to return a query response to mobile units that may be in transit. The cost that affects the most in centralized environments is the input/output. Communication cost is the most important in distributed environments. It is possible to formulate location dependent queries. There is difficulty in estimating the communication costs in distributed environments because the mobile host may be situated in different locations. There is a requirement of dynamic optimization strategies in the mobile distributed context.

Database Hoarding

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.

Database hoarding may be done at the application tier itself. The following figure shows a simple architecture in which a mobile device API directly retrieves the data from a database. It also shows another simple architecture in which a mobile device API directly retrieves the data from a database through a program, for ex: IBM DB2 Everyplace (DB2e).



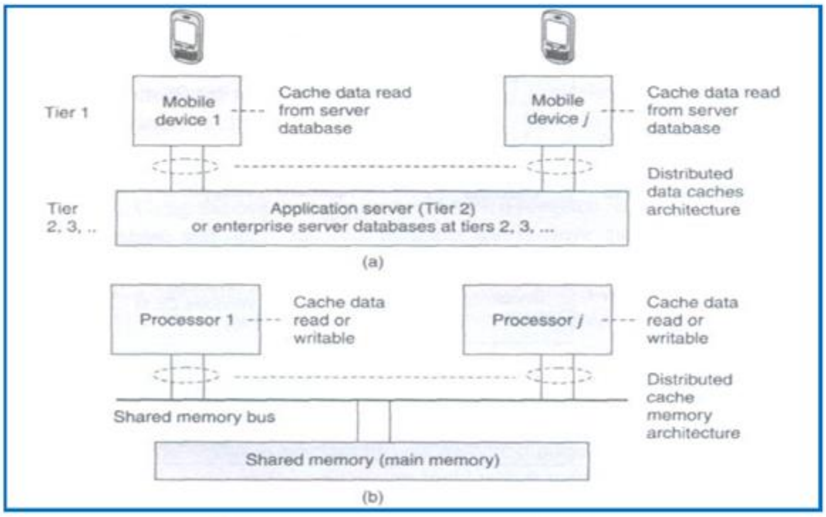
(a) API at mobile device sending queries and retrieving data from local database (Tier 1)
(b) API at mobile device retrieving data from database using DB2e (Tier 1)

Both the two architectures belong to the class of one-tier database architecture because the databases are specific to a mobile device, not meant to be distributed to multiple devices, not synchronized with the new updates, are stored at the device itself. Some examples are downloaded ringtones, music etc. **IBM DB2 Everyplace (DB2e)** is a relational database engine which has been designed to reside at the device. It supports J2ME and most mobile device operating systems. DB2e synchronizes with DB2 databases at the synchronization, application, or enterprise server.

The database architecture shown below is for two-tier or multi-tier databases. Here, the databases reside at the remote servers and the copies of these databases are cached at the client tiers. This is known as client-server computing architecture.

A cache is a list or database of items or records stored at the device. Databases are hoarded at the application or enterprise tier, where the database server uses business logic and connectivity for retrieving the data and then transmitting it to the device. The server provides and updates local copies of the database at each mobile device connected to it. The computing API at the mobile device (first tier) uses the cached local copy. At first tier (tier 1), the API uses the cached data records using the computing architecture as explained above. From tier 2 or tier 3, the server retrieves and transmits the data records to tier 1 using business logic and synchronizes the local copies at the device. These local copies function as device caches.

The advantage of hoarding is that there is no access latency (delay in retrieving the queried record from the server over wireless mobile networks). The client device API has instantaneous data access to hoarded or cached data. After a device caches the data distributed by the server, the data is hoarded at the device. The disadvantage of hoarding is that the consistency of the cached data with the database at the server needs to be maintained.



(a) Distributed data caches in mobile devices
(b) Similar architecture for a distributed cache memory in multiprocessor systems

Data Replication in Mobile Computing

Data Replication

Data Replication in mobile computing means the sharing of information to ensure data consistency between software and hardware resources connected via the internet, to improve reliability, availability, fault-tolerance, and accessibility of data.

In simpler terms, data replication is the process of storing different copies of the database at two or more sites in order to improve data availability in less time and at a cheaper cost.

Data replication in mobile computing is a popular fault tolerance technique for distributed databases.

Advantages of Data Replication

In modern mobile computing, scenario data replication has been adopted as an efficient way to ensure data availability, integrity, and an effective means to achieve fault tolerance. Data replication not only ensures the availability of the data but also minimize the communication cost, increase data sharing, and enhance the security of sensitive data. Data replication in mobile computing also determines when and which location to store the replica of data, controlling different data replicas over a network for efficient utilization of the network resources.

Data Replication Benefits

Important benefits of data replication are as below-

- **Reliability** − Data replication provides the reliability of data. In case of failure of any site, the database system continues to work since a copy is available at another site(s).
- **Reduction in Network Load** − since local copies of data are available through data replication. Therefore, query processing can be done with reduced network usage, particularly during prime hours.
- **Data updating can be done at non-prime hours** – Due to data replication data can be updated easily.
- **Quicker Response** − Availability of local copies of data ensures quick query processing and consequently quick response time.
- **Simpler Transactions** − Transactions require less number of joins of tables located at different sites and minimal coordination across the network. Thus, they become simpler in nature.

Disadvantages of Data Replication

- **Increased Storage Requirements** − Maintaining multiple copies of data is associated with increased storage costs. The storage space required is in multiples of the storage required for a centralized system.
- **Increased Cost and Complexity of Data Updating** − each time a data item is updated, the update needs to be reflected in all the copies of the data at the different sites. This requires complex synchronization techniques and protocols.

Goals of data replication

Data replication is performed with the purpose of

- Increasing the availability of data.
- Speeding up the query evaluation.

Types of data replication

There are two types of data replication

1. Synchronous Replication
In synchronous replication, the replica of the database is modified immediately after changes are made in the relation table.

So there is no difference between the original data and the replicated data table.

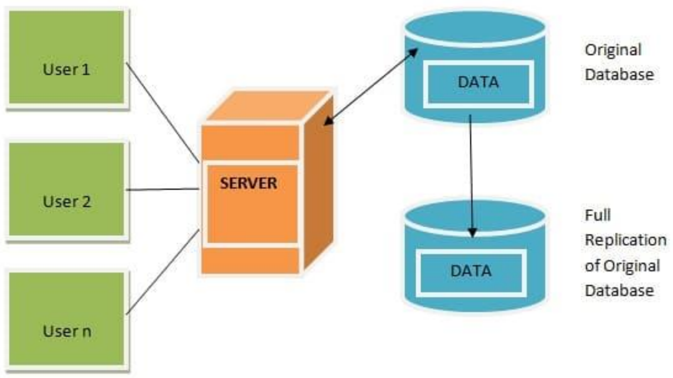
2. Asynchronous replication
In asynchronous replication, the replica will be modified after commit action is fired on to the database.

Replication Schemes

The three replication schemes are as follows:

1. **Full Replication scheme**

In full replication scheme, the database is available at all the locations to ease the user in the communication network



Advantages of full replication

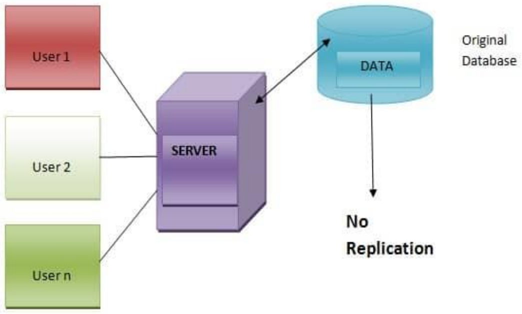
- It gives high availability of data. In this scheme, the database is available at each location.
- It supports faster execution of queries.

Disadvantages of full replication

- In a full replication scheme, concurrency control is difficult to achieve in full replication.
- During updating each and every side need to be updated therefore update operation is slower.

2. **No Replication**

No replication means each fragment is stored exactly at one location only.



Advantages of no replication

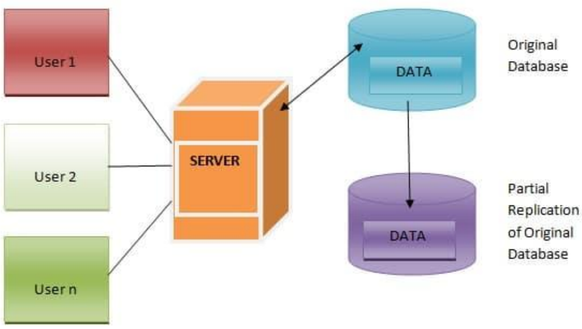
- Concurrency can be easily minimized.
- Easy recovery of data becomes easy.

Disadvantages of no replication

- Poor availability of data.
- Slows down the query execution process, because multiple clients are accessing the same data at the same server.

3. **Partial replication**

A partial replication scheme means only part of the or data fragments are replicated.



Adaptive clustering in mobile wireless networks

Personal communications and mobile computing require a wireless network infrastructure which is fast deployable, possibly multihop, and capable of multimedia service support. The first infrastructure of this type was the Packet Radio Network (PRNET), developed in the 70's to address the battlefield and disaster recovery communication requirements. PRNET was totally asynchronous and was based on a completely distributed architecture. It handled datagram traffic reasonably well, but did not offer efficient multimedia support. Recently, under the WAMIS (Wireless Adaptive Mobile Information Systems) and Glomo ARPA programs several mobile, multimedia, multihop (M3) wireless network architectures have been developed, which require some form of synchronous, time division infrastructure. The synchronous time frame leads to efficient multimedia support implementations. However, it introduces more complexity and is less robust in the face of mobility and channel fading. Clearly there are complexity vs performance tradeoffs in introducing various degrees of synchronization into the network. Figure 1, shows the cellular model commonly used in the wireless networks. A, B, C, and D are fixed base stations connected by a wired backbone. Nodes 1 through 8 are mobile nodes. A mobile node is only one hop away from a base station. Communications between two mobile nodes must be through fixed base stations and the wired backbone.

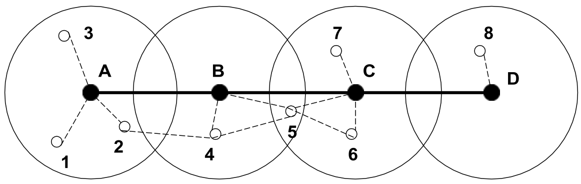


Figure 1: Conventional cellular networks (single-hop)

In parallel with (and separately from) the single hop cellular model, another type of model, based on radio to radio packet multihopping, has been emerging to serve a growing number of applications which rely on a fast deployable, wireless infrastructure. The classic examples are battlefield communications and (in the civilian sector) disaster recovery (fire, earthquake) and search and rescue. A recent addition to this set is the ‘‘adhoc’’ personal communications network, which could be rapidly deployed on a campus, for example, to support collaborative computing and access to the Internet during special events (concerts, festivals etc). Multihopping through wireless repeaters strategically located on campus permits to reduce battery power and to increase network capacity. More precisely, by carefully limiting the power of radios, we conserve battery power. Furthermore, we also cause less interference to other transmissions further away; this gives the additional benefit of ‘‘spatial reuse’’ of channel spectrum, thus increasing the capacity of the system. Interestingly, the multihop requirement may also arise in cellular networks. If a base station fails, a mobile node may not be able to access the wired network in a single hop. For example, in Figure 2, if base station B fails, node 4 must access base stations A or C through node 2 or node 5 which act as wireless multihop repeaters.

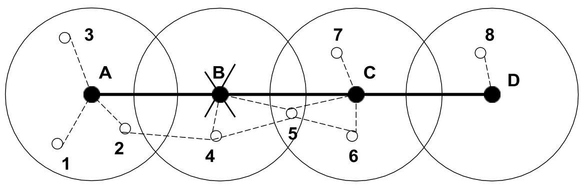


Figure 2: A multihop situation occurs when base station *B* fails.

In order to support multimedia traffic, the wireless network layer must guarantee QoS (band- width and delay) to real time traffic components. Our approach to provide QoS to multimedia consists of the following two steps: (a) partitioning of the multihop network into clusters, so that controlled, accountable bandwidth sharing can be accomplished in each cluster; (b) establishment of Virtual Circuits with QoS guarantee.

The objective of the clustering algorithm is to partition the network into several clusters. Optimal cluster size is dictated by the tradeoff between spatial reuse of the channel (which drivestoward small sizes), and delay minimization (which drives towards large sizes). Other constraints also apply, such as power consumption and geographical layout. Cluster size is controlled through the radio transmission power. For the cluster algorithm, we have so far assumed that transmission power is fixed and is uniform across the network.

Within each cluster, nodes can communicate with each other in at most two hops. The clusters can be constructed based on node ID. The following algorithm partitions the multihop network into some non-overlapping clusters. We make the following operational assumptions underlying the construction of the algorithm in a radio network. These assumptions are common to most radio data link protocols.

A1: Every node has a unique ID and knows the IDs of its 1-hop neighbors. This can be provided by a physical layer for mutual location and identification of radio nodes.

A2: A message sent by a node is received correctly within a finite time by all its 1-hop neighbors.

A3: Network topology does not change during the algorithm execution.

We can find from this algorithm that each node only broadcasts one *cluster* message before the algorithm stops, and the time complexity is $O(|V|)$ where V is the set of nodes. The clustering algorithm converges very rapidly. In the worst case, the convergence is linear in the total number of nodes. Consider the topology in Figure 3. After clustering, in Figure 4,we can find six clusters in the system, which are {1,2}, {3,4,11}, {5,6,7,8,9}, {10,12,13},{14,15,16,17}, {18,19,20}. To prove the correctness of the algorithm we have to show that: 1) every node eventually determines its cluster; 2) in a cluster, any two nodes are at most two hops away; 3) the algorithm terminates.

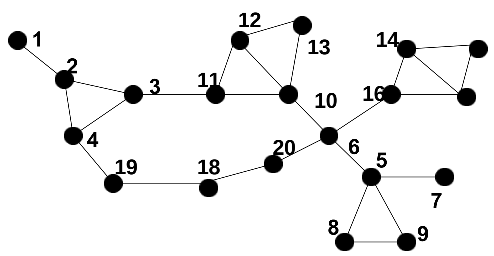


Figure 3: System topology

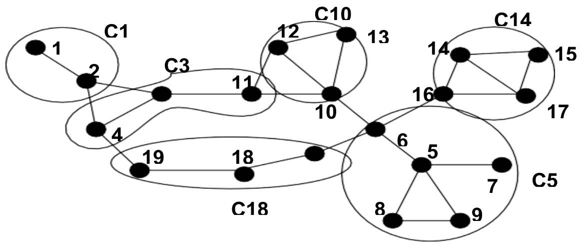


Figure 4: Clustering

Mobile File System

Designed for a mobile device, phone or smart card

• File system─ a basic middleware which glues applications to an OS

• Defined as a method of organizing and storing files on a storage device at computer or a mobile device

For the files, which may not be present on a single storage device

• Files may be distributed as in case of a distributed file system

• Present at different nodes in a network system

Consists of a master file which is the root directory

• Master file Stores all file headers

• A header contains description about a file

The second layer after the master file consists of dedicated files (directories) at the branches

• Dedicated file holds file groupings

• Each dedicated file may further have dedicated files and/or elementary files as branches

The elementary files are at third layer

• The elementary file holds the file header and the file data

Scalability (scalability in case of mobile file system means that the system should adjust the limits of file sizes and the number of files in the storage device as per the available memory capacity) . Support for defined semantics for sharing of files even in case of network failure

Support for disconnected operations and provision for reintegration of data from disconnected clients or server. High performance through client-side persistent caching

Provision for replication at server (Replication is defined as a process of repeating, making and offering a new copy of earlier ones)

• Replication by server means that server repeats and offers (broadcasts) the set of records using broadcast disk model) . Security, access control, authentication, and encryption

Continuous operation even in case of partial failure of network connectivity (Partial failure means disconnection between server and a few clients) . Network which adapts to the bandwidth available at a given instant

• An application performance improves if the file system adapts to the bandwidth variations