Unit VI

Chapter - 15 : Database Architecture

Q.1 What are the key elements of Parallel DB processing? Explain.

SPPU - May 18, Dec. 19, 4 Marks

Ans. : Parallel Databases System

Introduction

- A parallel database improves data processing speed by using multiple resources (CPU and Disk) in parallel.
- Parallel operations are becoming increasingly common to improve speed of operation and therefore study of Parallel databases is becoming more important.
- In organizations huge amount of data is handled, such huge amount of data needs high data transfer rate.
- Centralized and client server system is not much powerful for managing such applications.
- Parallel databases try to improve performance of database through parallelization of operations such Data loading, Query evaluation etc.
- We can use thousands of small processors for making a parallel machine.

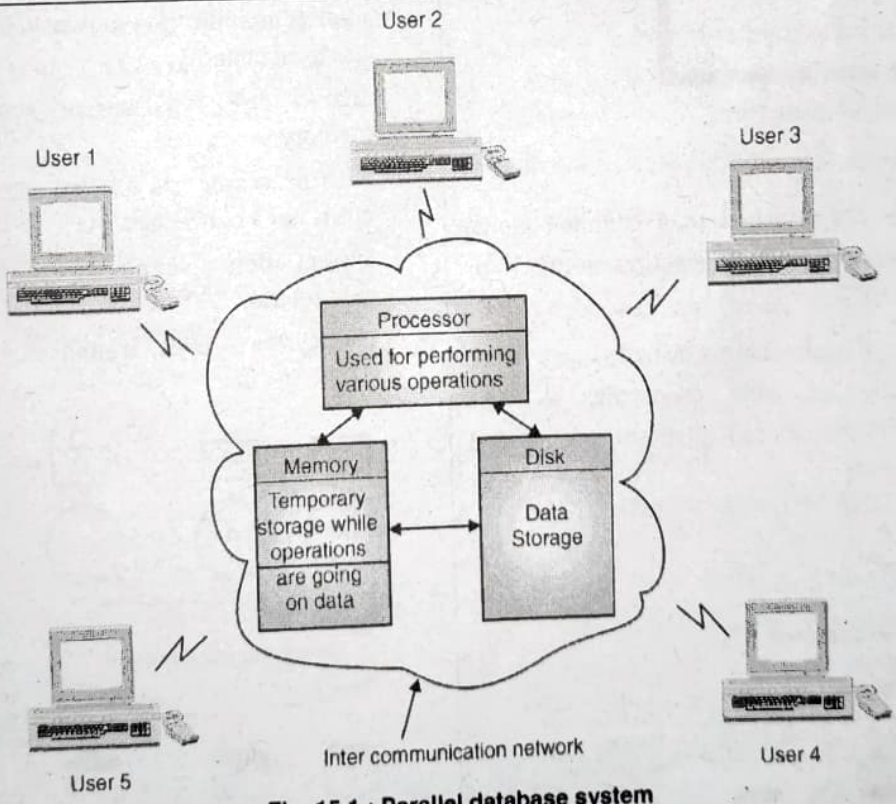

Fig. 15.1 : Parallel database system

Goals of Parallel Databases

(a) Improved performance

Using multiple resources (e.g., CPUs and disks) in parallel we can significantly improve performance of system.

(b) Increased availability

If a site containing a relation (table in database) is not available, then the relation continues to be available from another site which has a copy of that data.

(c) Increased reliability

If a site containing a relation (table in database) fails to work, the relation continues to be available from another site which has a copy of that data. Hence this leads to more reliable system.

(d) Distributed data access

An organization may need to access data which belongs to different sites, as it may be possible to have multiple branches of a company.

Q.2 Explain Architecture of Parallel Databases.

SPPU - Dec. 17, May 18, 6 Marks

Ans. : Architecture of Parallel Databases

Parallelism in databases represents one of the most successful instances of parallel computing system.

Types of Parallel Databases

1. Shared Memory System
2. Shared Disk System
3. Shared Nothing System

(A) Shared Memory System

(a) Architecture details

- Multiple CPUs are attached to a common global shared memory via interconnection network or communication bus.
- Shared memory architectures usually have large memory caches at each processor, so that referencing of the shared memory is avoided whenever possible.

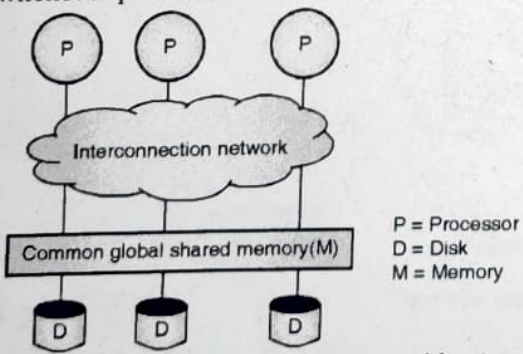

Fig. 15.1 : Shared memory system architecture

- Moreover, caches need to be coherent. That means if a processor performs a write to a memory location, the data in that memory location should be either updated at or removed cached data.

(b) Advantages

- Efficient communication between processors.
- Data can be accessed by any processor without being moved from one place to other.
- A processor can send messages to other processors much faster using memory writes.

(c) Disadvantages

- Bandwidth problem
- Not scalable beyond 32 or 64 processors, since the bus or interconnection network will get into a bottleneck.
- More number of processors can increase waiting time of processors.

(B) Shared Disk System

(a) Architecture details

- Multiple processors can access all disk directly via inter communication network. But, every processor has local memory.
- Shared disk has two advantages over shared memory.
- Each processor has its own memory; the memory bus is not a bottleneck.
- System offers a simple way to provide a degree of fault tolerance.
- The systems built around this architecture are called clusters.

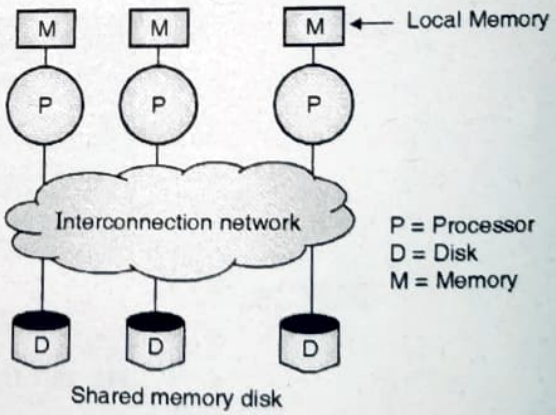

Fig. 15.2 : Shared memory disk architecture

(b) Advantages

- Each CPU or processor has its own local memory, so the memory bus will not face bottleneck.
- High degree of fault tolerance is achieved.

- Fault tolerance : If a processor (or its memory) fails, the other processor can take over its tasks, since the database is present on disks and are accessible to all processors.
- If one processor fails, other processors can take over its tasks, since database is on shared disk that can be accessible from all processors.

(c) Disadvantages

- Some memory load is added to each processor.
- **Limited scalability :** Not scalable beyond certain point. The shared-disk architecture faces this problem because large amounts of data are shipped through the interconnection network. So now the interconnection to the disk subsystem is a bottleneck.
- The basic problem with the shared-memory and shared-disk architecture is interference. As more CPUs are added, existing CPUs are slowed down because of the increased contention for memory accesses and network bandwidth.

(d) Applications

Digital equipment corporation : DEC cluster running Relational Databases were one of the early commercial user of shared disk database architecture. Now this is owned by Oracle.

(C) Shared Nothing Disk System

(a) Architecture details

- Each processor has its own local memory and local disk.
- A processor at one node may communicate with another processor using high speed communication network.

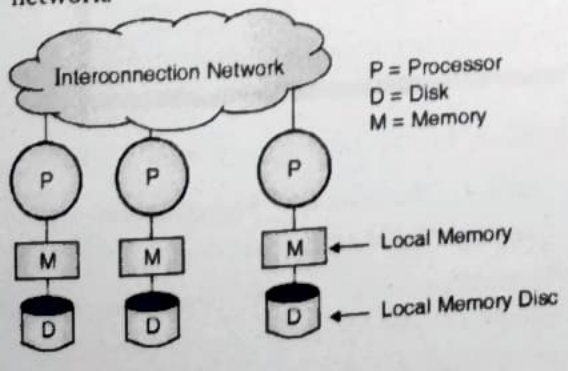

Fig. 15.3 : Shared nothing architecture

- Any terminal can act as a Node which functions as Server for data that is stored on local disk.
- Moreover, the interconnection networks for shared nothing systems are usually designed to be scalable, so that we can increase transmission capacity as more nodes are added to the network.

(b) Advantages

- In this type of architecture no need to go through all I/O, Only a single interconnection network queries which access non local disk can pass through the network.
- We can achieve High degree of parallelism. i.e. Number of CPU and disk can be connected as desired.
- Shared nothing architecture systems are more scalable and can easily support a large number of processors.

(c) Disadvantages

- Cost of communication and of non local disk access is higher than other two architectures since sending data involves software interaction at both ends.
- Requires rigid data partitioning.

(d) Applications :

- The teradata database machine uses shared nothing database architecture.
- Grace and the Gamma research prototypes.

(D) Hierarchical System

Architecture details

- The **hierarchical architecture** comes with combined characteristics of shared memory, shared disk and shred nothing architectures.
- At the top level, the system consists of nodes connected by an interconnection network and they do not share disks or memory with one another.
- This architecture is attempts to reduce the complexity of programming such systems yields to **distributed virtual-memory** architectures, where logically there is a single shared memory, the memory mapping hardware coupled with system software, allows each processor to view the disjoint memories as a single virtual memory.
- The **hierarchical architecture** is also referred to as **non uniform memory architecture**.

Q.3 Explain Data Replication and Data Fragmentation in Distributed Data Storage.

SPPU - May 18, Dec.18, May 19, Dec. 19, 6 Marks

Ans. : Data Fragmentation / Distributed Database Design

Introduction

- The process of decomposing the database into smaller multiple units called as **fragments.**
- These fragments may be stored at various sites, is called **data fragmentation.**

Completeness constraint :

The most important condition of data fragmentation process is that it must be complete i.e. once a database is fragmented, it must be always possible to reconstruct the original database from the fragments.

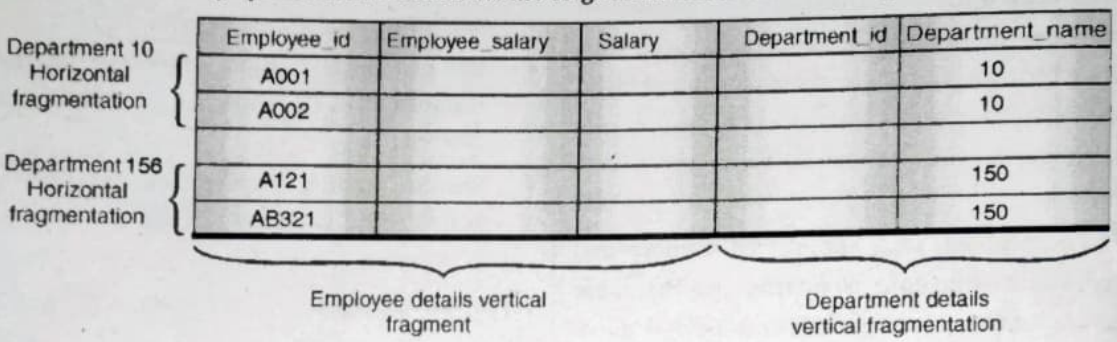

Fig. 15.4 : Horizontal and vertical fragmentation

Fragmentation Schema

It is a set of fragments that includes all attributes and tuples in the database and satisfies the condition that the whole database can be reconstructed from the fragments by applying some sequence of database operations.

Types of Data Fragmentation

(a) Horizontal data fragmentation
(b) Vertical data fragmentation
(c) Mixed data fragmentation

(A) Horizontal Fragmentation

(1) Introduction

- Horizontal fragmentation divides a relation horizontally into group of rows (tuple) to create subsets of tuples specified by a condition on one or more attributes of relation.
- The tuples that belong to the horizontal fragment is specified by some condition on one or more attributes of the relation.

(2) Overview

- Horizontal fragmentation is group of rows in relation.
- Horizontal fragments are specified by the 'SELECT' operation of the relational algebra on single or multiple attributes.

Example :

Select all students of computer branch.

$\sigma_{Branch = "COMP"}$ (students)

(3) Types

(a) Primary horizontal fragmentation
(b) Derived horizontal fragmentation
(c) Complete horizontal fragmentation

(a) Primary horizontal fragmentation

- Primary horizontal fragmentation is the fragmentation of primary relation.
- Relation on which other relations are dependent using foreign key is called as primary relation.

Example :

Partition 1 : All employees belong to department number 10.

$R_1 \leftarrow \sigma_{Dept = 10}$ (EMP)

Partition 2 : All employees belong to department number 20.

$R_2 \leftarrow \sigma_{Dept = 20}$ (EMP)

(b) Derived horizontal fragmentation

- Horizontal fragmentation of a primary relation introduces Derived horizontal fragmentation of other secondary relations that are dependent on primary relations.
- The fragmentation on some other fragmentation is called as Derived horizontal fragmentation.

Example :

Partition 1 : All employees belong to department number 10 and having age above 25.

$\sigma_{Age > 25}$ (R_1)

Partition 2 : All employees belong to department number 20 and having age above 35.

$\sigma_{(Age > 35)}$ (R_2)

(c) Complete horizontal fragmentation

- It generates a set of horizontal fragments that include each and every tuple of original relation.
- For reconstruction of relation completeness is required. As every tuple must belongs to at least one of the partition.
- Consider relation below as R now subdivided in P_1, P_2, P_3 and P_4 In case of complete horizontal fragment if relation R contains 50 tuples than total number of tuples in below 4 partitions should be 50 or more than 50.

Example :

Partition 1 : All employees belong to department number less than 20.

$R_1 \leftarrow \sigma_{DeptNo <= 20}$ (EMP)

Partition 2 : All employees belong to department number less than 30.

$R_2 \leftarrow \sigma_{DeptNo = 30}$(EMP)

Partition 3 : All employees belong to department number less than 40.

$R_3 \leftarrow \sigma_{DeptNo = 40}$ (EMP)

Partition 4 : All employees belong to department number above 40.

$R_4 \leftarrow \sigma_{Dept > 40}$(EMP)

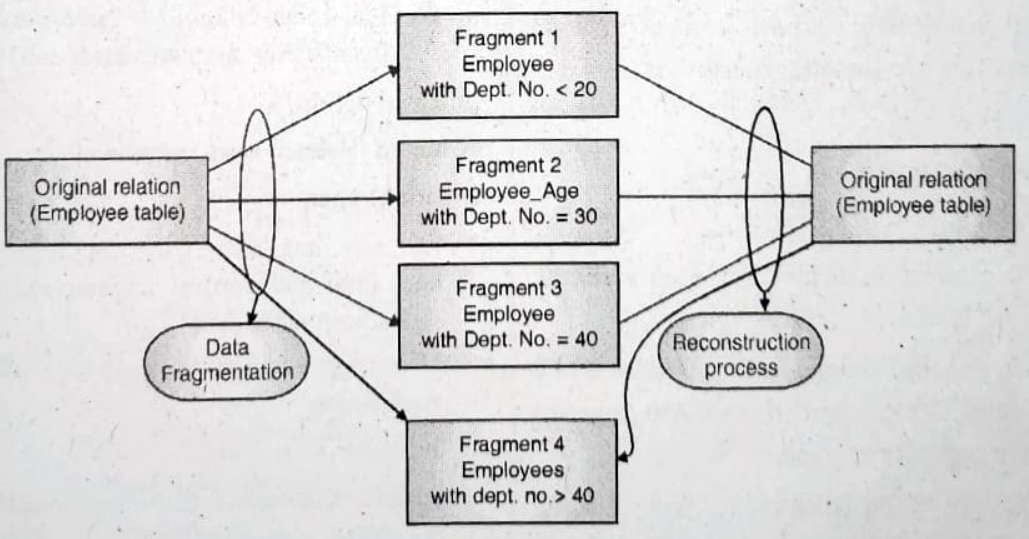

Fig. 15.5 : Complete horizontal fragmentation

(d) Disjoint horizontal fragmentation

- It generates a set of horizontal fragments where no two fragments have common tuples.
- That means every tuple of relation belongs to one and only one fragment.

Example :

Partition 1 : All employees having Age 18 or less.

$R_1 \leftarrow \sigma_{EmpAge <= 18}(EMP)$

Partition 2 : All employees having Age above 18 and below 65.

$R_2 \leftarrow \sigma_{EmpAge > 18 \text{ AND } EmpAge < 65}$ (EMP)

Partition 3 : All employees having Age above 65.

$R_3 \leftarrow \sigma_{EmpAge >= 65}$ (EMP)

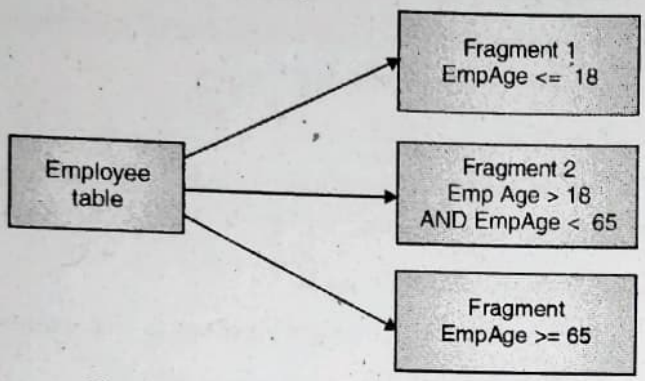

Fig. 15.6 : Disjoint horizontal partitioning

(4) Reconstruction process for horizontal fragments

- To reconstruct the original relation we need to perform set UNION (U) operation on fragments.
- The original relation can be reconstructed if and only if completeness constraint is satisfied.
- **Example :** Consider relation shown in complete horizontal partitioning we can reconstruct relation as it is satisfying completeness constraints

$R \leftarrow R_1 \cup R_2 \cup R_3 \cup R_4$

(B) Vertical Fragmentation

(1) Introduction

- Vertical fragmentation divides a relation vertically into group of columns.
- When each site does not need all the attributes of a relation, vertical fragmentation is used to fragment the relation vertically by columns.
- It is necessary to include primary key or some common candidate key in every vertical fragment to reconstruct the original relation from the fragments.

(2) Overview

- Vertical fragmentation is group of columns in relation.
- Vertical fragmentation can be specified by 'PROJECT' operation of the relational algebra.

Example :

Select Name and address of all students of computer branch.

$\pi_{Name, Address}$ (students)

(3) Types of vertical fragmentation

- Complete vertical fragmentation
- It generates a set of vertical fragments that include all the attributes of original relation and share only primary key of original relation.
- Consider relation with following schema
- Student (sid, Name, Age, Address, Phone, class, fees)
- $P_1 \rightarrow \pi_{Sid, Name, Address}$ (students)
- $P_2 \rightarrow \pi_{Sid, Age, Phone}$ (students)
- $P_3 \rightarrow \pi_{Sid, class, Fees}$ (students)

(4) Reconstruction

- To reconstruct the original relation we need to perform FULL OUTER JOIN ($\bowtie$) operation on fragments.
- The original relation can be reconstructed if and only if completeness constraint is satisfied that means there should be either one column which is common between two partitions.

Example :

Consider above relation we can reconstruct relation as it is satisfying completeness constraints

$R \leftarrow P_1 \bowtie P_2 \bowtie P_3$

(C) Mixed (Hybrid) Fragmentation

(1) Introduction

- We can mix two types of fragmentation i.e. horizontal and vertical fragmentations yielding a mixed fragmentation.
- This fragmentation is generally used in many applications.

(2) Overview

- Mixed fragmentation is group of columns and rows in relation.
- Mixed fragmentation can be specified by 'PROJECT' and 'SELECT' operation of the relational algebra.

(3) Example

- Mixed fragmentation can be applied to student table for following student schema;

Student table

Sid	SName	age	Branch id	Bname

- Student table contains information about student and branches associated with particular student. Branches can be Computer Science (CS) or IT branch.

Table : Student database

	SName	Age	Sid	Branchid	Bname	
$F_1 \rightarrow$	Smriti	20	1	10	CS	$\leftarrow F_3$
	Jay	24	2	10	CS	
	Ashok	22	3	10	CS	
	Neha	21	4	20	IT	
	Raj	20	5	20	IT	$\leftarrow F_4$
$F_2 \rightarrow$	Harshad	23	6	20	IT	

(a) Fragment 1 : Student details of all students in 'CS' Branch.

$F_1 \rightarrow \pi_{Sid, SName, age} (\sigma_{Bname='CS'}(student))$

Sid	SName	Age
1	Smriti	20
2	Jay	24
3	Ashok	22

(b) Fragment 2 : Student details of all students in 'IT' Branch.

$F_2 \rightarrow \pi_{Sid, SName, age} (\sigma_{Bname='IT'}(student))$

Sid	SName	Age
4	Neha	21
5	Raj	20
6	Harshad	23

(c) Fragment 3 : Find all student's branch details of CS Branch

$F_3 \rightarrow \pi_{Sid, Branchid, BName} (\sigma_{Bname = 'CS'}(student))$

Sid	Branchid	Branch
1	10	CS
2	10	CS
3	10	CS

(d) Fragment 4 : Find all student's branch details of IT Branch.

$F_4 \rightarrow \pi_{Sid, Branchid, BName} (\sigma_{Bname='IT'}(student))$

Sid	Branchid	Branch
4	20	IT
5	20	IT
6	20	IT

(4) Reconstruction

- To reconstruct the original relation by performing Union and FULL OUTER JOIN ($\bowtie$) operation in appropriate order on fragments.
- The original relation can be reconstructed if and only if completeness constraint is satisfied that means there should be either one column which is common between two partitions.

Example :

To reconstruct above fragmentation we will first find union of F_1 and F_2 which will give me details of all students.

$F_1 \cup F_2$

Sid	SName	Age
1	Smriti	20
2	Jay	24
3	Ashok	22
4	Neha	21
5	Raj	20
6	Harshad	23

Now, we will find details of or department in which student study by taking Union of F_3 and F_4.

$F_3 \cup F_4$

Sid	Branchid	Bname
1	10	CS
2	10	CS
3	10	CS
4	20	IT
5	20	IT
6	20	IT

Now to reconstruct main table we Join above two tables using Join ($\bowtie$) with help of 'Sid' as common column.

$$R \Rightarrow (F_1 \cup F_2) \bowtie (F_3 \cup F_4)$$

Above query will returns the original relation as in table: student database.

□□□

Chapter - 16 : NoSQL Database

Q.1 Write a short note on Internet database.

SPPU - May 18, May 19, 4 Marks

Ans. : Internet Database

1. Introduction

- Large amount of data can be easily managed by database technology.
- The internet / web is a good way to present information to the user.
- In internet databases data management is separated from presentation.
- This improves the efficiency of the application. Updating data, finding information becomes very easy.

  Example : Employee database, airline / railway booking, e-commerce etc.
- Web-To-Database Middleware (ColdFusion)

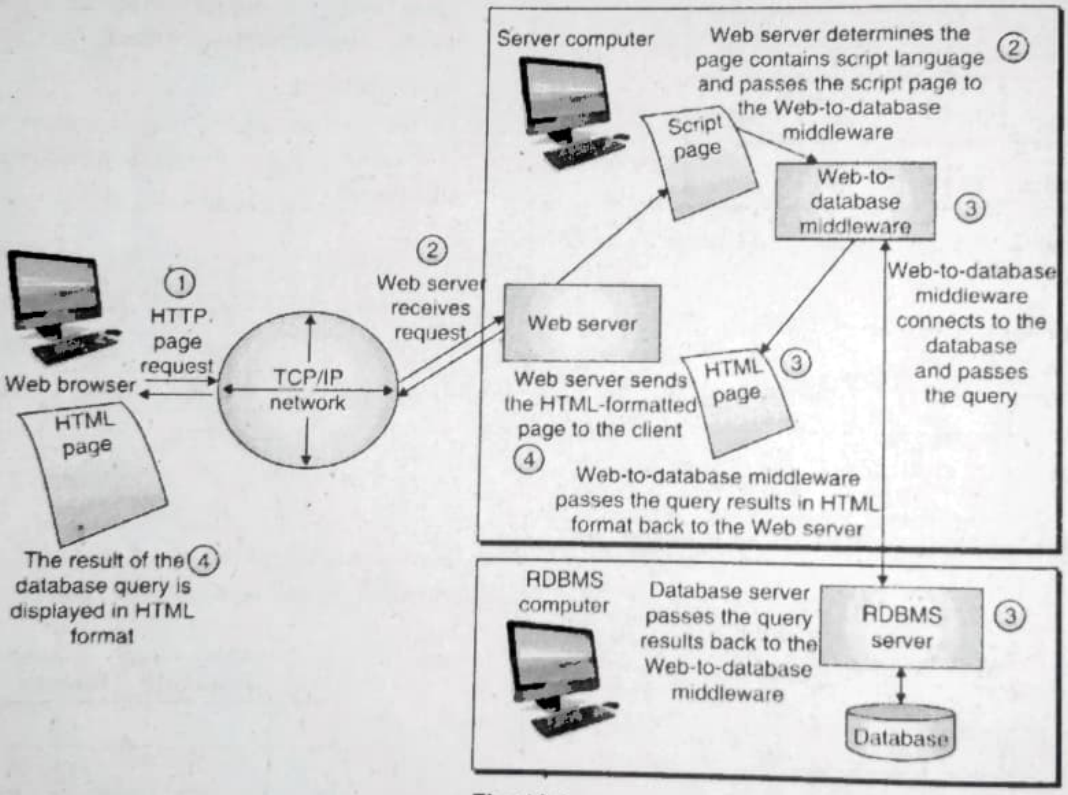

Fig. 16.1

- Cold fusion application middleware helps us in following ways :
  - Through web pages we can connect to a database easily.
  - Data can be represented on the web page in any format.
  - Dynamic web pages creation, updation becomes very easy.
  - Relationships, referential integrity etc. can be easily defined.
  - Database fields can be easily created or deleted.
  - Following Fig. 16.2 shows thathow cold fusion works in internet databases.

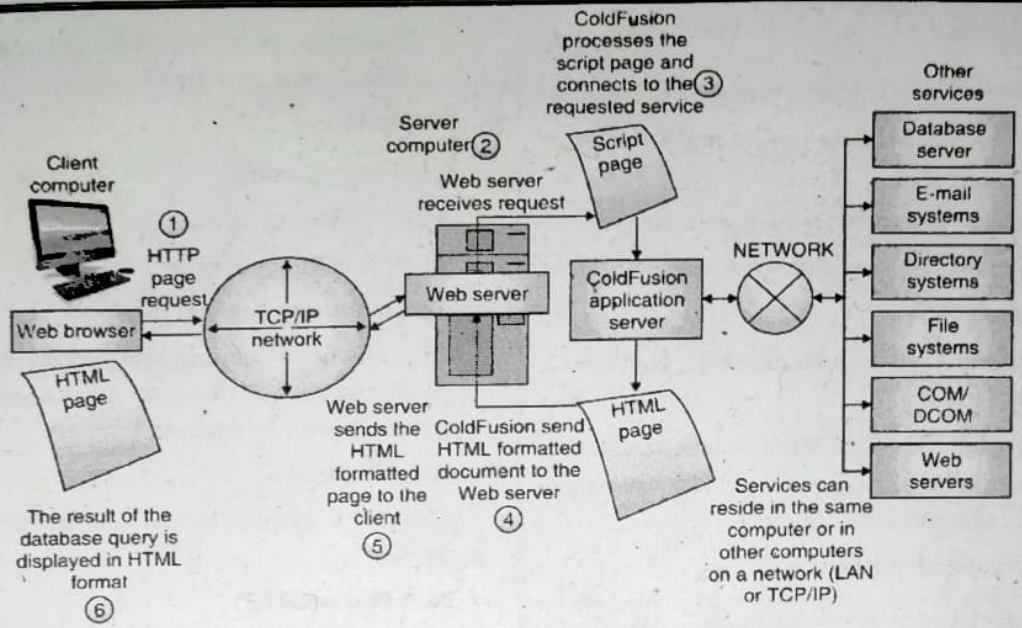

Fig. 16.2 : How cold fusion works ?

2. Web Server Interfaces

- **Common Gateway Interface (CGI)**

  Scripts are nothing but small programs written in programming languages link C, VB, C++. Script files are used by CGI to perform a specific function. Client passes parameters to the webservers.

- **Application Programming Interfaces (APIs)**

  APIs Web server interfaces are much more efficient and faster than CGI scripts. APIs are implemented as shared code or as dynamic-link libraries (DLLs). Thw following Fig. 16.2.3 shows API and CGI web Server Interfaces.

  The API And CGI Web Server Interfaces.

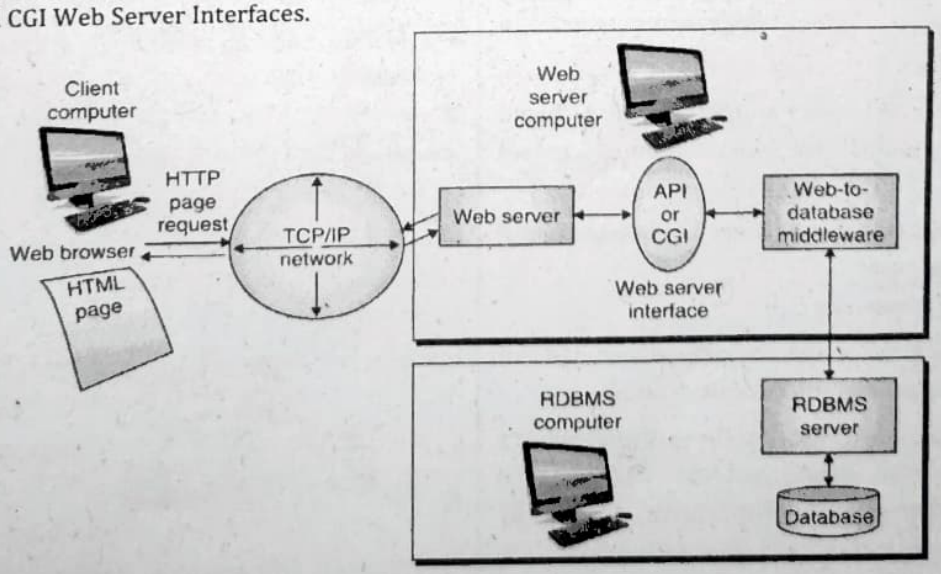

Fig. 16.3

3. Advantages

- Data is available all the time irrespective of location.
- Scalability is good i.e. data can be easily added or deleted.
- Saves hardware and software management cost.
- Huge amount of data is available for searching.

4. Disadvantages

- Secuiry issue : As data is online there is always a security threat.
- As data is available in different formats, at different locations, switching from one database technology to another increases cost.
- Network failure, server failure.

Q.2 Write a short note on Mobile database.

SPPU - May 18, Dec. 19, 6 Marks

Ans. : Mobile Databases

1. Introduction

Recent developments in wireless networking and introduction of portable devices have given rise to new era of mobile computing.

2. Goals

(a) Mobility

Mobile computing allows user or client to access required information from anywhere and at any point of time.

(b) Battery life

- Mobile devices works on battery which should have unlimited life (or unlimited power supply).
- Hence limited battery power should not delimit mobility of user.

(c) Changing networks

- Changing topology of the network should not cause any problems to mobile devices.
- In mobile environment, these problems are more difficult, mainly because of the limited and discontinuous connectivity offered by wireless networks.

(d) Some of the software issues which may involve data management problem, transaction management or database recovery are also present in distributed database systems

3. Examples :

(a) News reporting

(b) E- brokering services

(c) Mobile Billing services

Mobile Computing Architecture

1. In a mobile computing architecture made of fixed hosts and base stations, are interconnected through a high-speed wired network.

2. Fixed Hosts (FH)

Computers that can be configured to manage mobile units.

3. Base Stations (BS)

- Working as gateways for the mobile units.
- They are ready with wireless interfaces for mobile units or clients.

4. Cell (C)

- To manage the mobility of units, the entire coverage area is divided into one or smaller domains called as cell.
- Each cell has at least one base station.

5. Mobile Units (MU)

- Users or Clients for base station in mobile network
- Mobile units can move freely within cell or between cells.

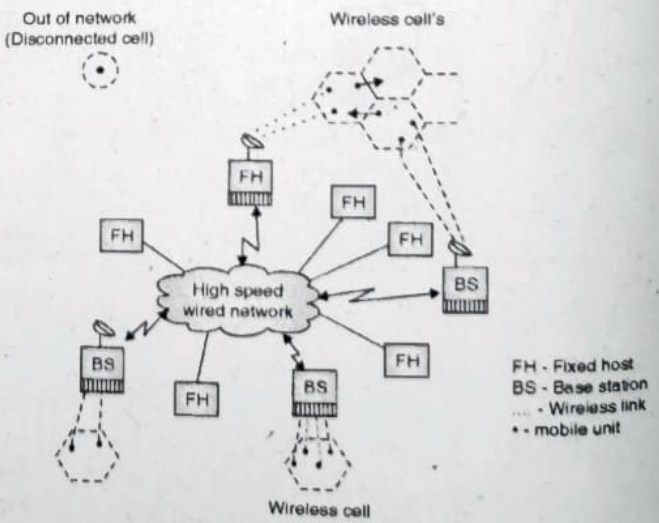

Fig. 16.1 : Mobile computing architecture

Characteristics of Mobile Environments

Mobile devices have some special characteristics that must be taken into account when designing applications for such environments.

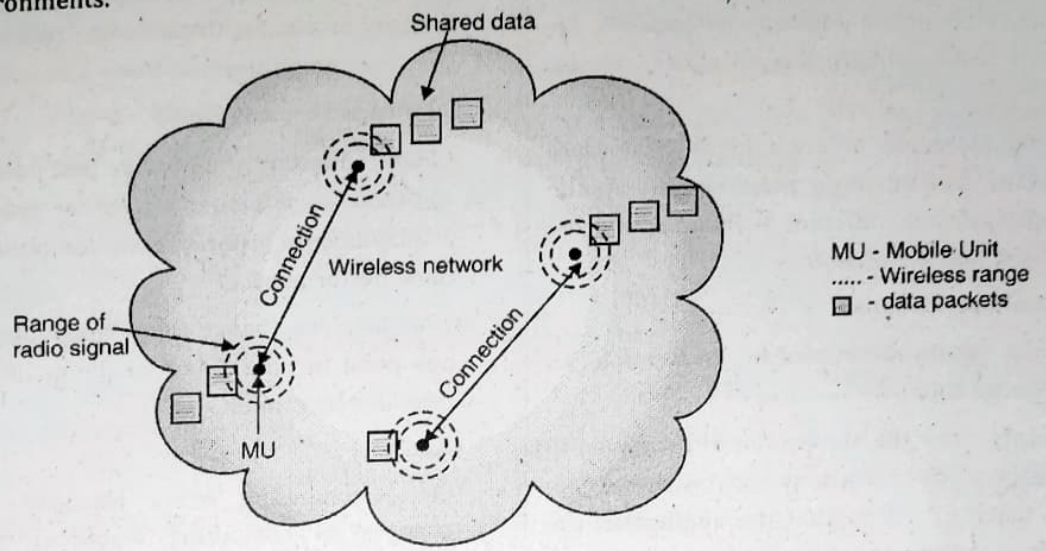

Fig. 16.2 : Mobile environment

1. Battery

- Applications must be designed to be as energy-efficient as possible due to the limited battery power of mobile devices.
- The mobile applications should therefore be designed to stop processes when idle and save CPU usage when possible.
- Battery life is directly related to battery size, and indirectly related to the mobile device's capabilities.
- Caching data can also reduce power consumption by eliminating the need to make energy consuming wireless data transmissions for each data access.

2. Data rate

- The wireless mediums on which mobile units and base stations communicate have bandwidths significantly lower than those of a wired network.
- Hence the data rate of mobile systems is generally low.

3. Client connectivity

- Applications must be designed to provide maximum mobility to mobile device or user.
- Applications are designed in wireless environment, have lower data rate and limited connectivity.
- Intermittent connectivity can be intentional or unintentional.
- Unintentional disconnections happen in areas where wireless signals cannot reach i.e. no coverage area.
- Intentional disconnections occur by user intent e.g., during an airplane takeoff, or when the mobile device is powered off.

4. Server connectivity

- Servers must keep track of client locations in order to efficiently send messages to them.
- Client data should be stored in the network location that minimizes the traffic.

5. Memory

- Applications must be designed for environments that have less storage space and dynamic memory than desktop environments; this can be done using memory-efficient data structures and reuse of code.
- Applications must also be robust, since users generally find applications that crash easily, very tedious to use, especially so because devices are usually always on.

6. User interface

- Applications must be designed for environments that have small screens and input methods that differ from those of the desktop environment, for example the 12-key numeric keypad, stylus, and soft keys.
- In addition, there are different kinds of devices, which means that attention must be paid to the various display sizes, different keyboards, and the different look and feel of devices.

7. Execution of applications

- Applications can be interrupted by, for example, an SMS or a phone call.
- These events cause the application to be moved to the background. For situations such as these, care must be taken to ensure that the application can recover from the situation smoothly.

Q.3 Write a short note on Cloud database.

SPPU - May 18, May 19, Dec.19, 6 Marks

Ans. : Cloud Database

What is Cloud?

- Cloud refers to a network, internet or communication medium through which we can access computing resource located at remote location.
- Cloud can contain services like email, storage and web conferencing etc.

Cloud Features

1. On demand services / self service

- On-demand service refers to the service provided by cloud computing vendor enables user use cloud resources on demand whenever they actually need it.
- In on-demand self-service, the user makes use of online control panel to accesses cloud services.
- It is possible to upgrade or downgrade the cloud services as organisation needs it without investing lot of extra cost.

2. Pay per use pricing model

- Cloud Computing is affordable for start-up or small scale organisations to use infrastructure and services at lower cost without investing lot of upfront setup cost.
- User will pay only for computing resources which is actually in use. So, Organisation requires paying as they use computing resources.

3. Quality of Services (QoS)

- Cloud computing will provide best possible services to users. Cloud computing service guarantees 24x7 availability, provision of extra computing resources, Good performance.
- Customer can change cloud computing platform at any point of time based on the quality of services provided by vendor.

4. Elasticity

- Cloud computing offers elasticity in adopting changes in computing resources as workload changes by acquiring new resources and releasing extra resources in an autonomic manner.
- In general for cloud computing or for cloud application Cost, Quality, and Resources are basic parameters which require elasticity.

5. Customization

There are thousands of users are using cloud at Sam point of time. This environment requires lot of customization for every user for giving data security and simplicity for accessing data.

Q.4 Write a short note on SQLite database.

SPPU -Dec.17 , May 18, Dec. 19, 4 Marks

Ans. : SQLite Database

- SQLite is a free open souce RDBMS engine.
- SQLite is a self-contained, serverless, zero-configuration and transactional SQL embedded RDBMS engine.
- SQLite implements most of the SQL-92 standard for SQL.
- The SQLite library is small. It takes less than 500 KB space. Moreover SQLite database is a single file which we can easily store anywhere in the system.
- SQLite file is platform independent and can be easily portable. It supports many operating systems irrespective of their architectures i.e. 32 bit or 64 bit.

- C programming language is used to develop SQLite. It has support for many different and popular languages like Java, C++, C#, Python, VB, TCl etc.
- As it is an open source technology the source code of SQLite is easily available for public.
- It is very useful in memory constrained gadgets such as cell phones, PDAs, and MP3 players.
- Transactions are ACID even if interrupted by system crashes or power failures.

Features of SQLite

- Ensures ACID properties even after system crashes and power failures.
- No setup or administration needed.
- Small size.
- Portable and platform independent.
- Open source.
- Full SQL implementation with advanced features.
- Support for terabyte-sized databases and gigabyte-sized data.
- Easy and simple to use API.
- No external dependencies hence self contained.
- Readable code and available in public domain.

Applications of SQLite

Embedded devices and the internet of things

With no administration, SQLite can be easily used with cell phones, game consoles, televisions, set top boxes, watches, remote sensors, robots, automobiles, home appliancesetc.

Application file format

It is a cross platform file. It can be used on various operating systems, both 32 and 64 bit architectures.

Websites

SQLite works great with websites having low to medium traffic.

Data analysis

With full SQL implementation it can be used to analyze large datasets. Tcl or python can be used for more complex analysis.

Education and Training

Simple and easy to install and use. Open source, Very small in size. Hence very useful for education and training.

Prototyping and extensions

As it is lightweight, platform independent it is good for experimenting.

Good file archives

SQL lite can be used as a substitute for ZIP archives or Tarballs.

Q.5 Explain in brief the advantage of Mongo DB over RDBMS. **SPPU –Dec. 17, May 18, Dec. 18, 6 Marks**

Ans. : MongoDB

- MongoDB is an open-source document type database.
- MongoDB is implemented in C++.
- High performance, high availability and automatic scaling are the important features. MongoDB has its own ad-hoc query language with rich features set.
- Large numbers of use cases can be easily handled.
- Mobile applications, CMS (Content Management System), E-commerce, Gaming applications, Analytics, Archiving and Logging are the applications areas of MongoDB.
- Multi-document transactions are not possible in MongoDB.
- MongoDB does provide atomic operations on a single document.
- The biggest advantage of MongoDB is that, as a cache memory it automatically uses all free memory available on the machine.

Collection and Document

- Collection is equivalent to a table in RDBMS. A collection is a group of documents which exists within a single database. Collection is schemaless.
- Different fields can be created with different documents in a collection. But typically, all the documents in a single collection are of similar or related purpose.

- A document is a set of key value pairs. Documents have dynamic schema.

RDBMS terminology with MongoDB terminology

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by mongodb itself)
Database Server and Client	
Mysqld/Oracle	Mongod
mysql/sqlplus	Mongo

Document Database

In MongoDB document is used to store data. Each document consists of fields and values. MongoDB documents and JSON objects are similar to each other. Other documents, arrays etc. can be included in the field values.

Documents have following advantages :

- Due to embedded documents and arrays, joins are avoided. Hence less expensive.
- Document uses dynamic schema.
- In popular programming languages documents are native data types.
- A MongoDB document is as follows,

```
{   movie : "Hanuman",
    Ticket : 100,
    Screen : 02,
}
```

MongoDB Schema

MongoDB uses dynamic schemas. Documents in a collection may have different set of fields. Due to this polymorphism can be easily used. Without defining the structure, i.e. the fields or the data types of the documents we can create a structure. Adding new field and deleting existing fileds can be easily done.