1) What is distributed system? What are it's objectives? Differentiate between centralized and distributed system.

=> A distributed system is a collection of independent computers that appears to it's users as a single coherent system.

This definition has several important aspects.

- The first one is that a distributed system consists of components that are autonomous.
- A second aspect is that users think they are dealing with a single system. This means that one way or the other the autonomous components need to collaborate.

Some of the objectives of distributed system are :-

a) Cost
Better price/performance as long as commodity hardware is used for the component computers.

b) Reliability
By having redundant components the impact of hardware and software faults on users can be reduced.

c) Inherent distribution
The ~~end~~ Naturally and physically distributed.

d) Transparency
The end users can be concealed or hidden from the actual separation of the distributed system so that the user feels that everything is transparent to everyone.

e) Scalability
Resources such as processing and storage capacity can be increased significantly.

1) Dependability
The dependence of a system on another system can be acheived to solve a particular task jointly.

2) Performance
By using the combined processing and storage capacity of many nodes, performance levels can be reached that are beyond the range of centralized machines.

4) Flexibility
Easily can be added or removed a node.

The differences between centralized and distributed system are as follows:

Centralized System	Distributed System
1) Allow better usage of all data and applications are stand	suits specific needs of all units within organization.
ardized throughout.	
2) Organization better controlled and managed better security as all data and processing tasks are performed and kept at the central location.	Independence: no harm to other unit if one fails.
3) No data or application redundancy as whole system is managed by the main computer at central location.	Difficulties in communication and data exchange between units due to nonstandarized systems and data design.
4) Dependence: If the main system fails, everything stops.	Redundancy of data and applications are likely. As all data are distributed, data security is major issues.

2) Explain some of the characteris
- tics of distributed system.

=> A distributed system is a colle
- ction of independent computers that appears to it's users as a single coherent system.

Some of the characteristics of distributed system are as follows:

a) Resource sharing
It means that existing resources in a distributed system can be accessed or remotely accessed across multiple computers in the system. Computers in distri
- buted system share resources like hardware (disks and printers), software (files, windows and data objects) and data.

b) Heterogeneity
In distributed systems components can have variety and differences in networks; computer hardware, operating systems, programming languages and implementations by different developers.

c) Openness
Openness is concerned with exten
- sions and improvements of distributed systems. The distri
- buted system must be open in terms of hardware and software. In order to make a distributed system open,

1) A detailed and well-defined inter
- face of components must be public
- hed.

2) Should standardize the interfaces of components.

3) The new component must be easily integrated with existing comp
- onents.

d) Concurrency
Concurrency is a property of a system representing the fact that multiple activities are

executed at the same time. The concurrent execution of activities takes place in different components running on multiple machines as part of a distributed system. In addition, these activities may perform some kind of interactions among them. Concurrency reduces the latency and increases the throughput of distributed system.

e) Scalability
It is mainly concerned about how the distributed system handles the growth as the number of users for the system increases. Mostly we scale the distributed system by adding more computers in the network components should not need to be changed when we scale the system. Components should be designed in such a way that is scalable.

f) Fault tolerance
In distributed system, software, hardware, network anything can fail. The system must be designed in such a way that is available all the time even after something has failed.

g) Transparency
Distributed systems should be perceived by users and application programmers as a whole rather than a collection of cooperating components. Transparency can be of various types of like access, location, concurrency, replication etc.

3) Explain different types of distributed system.
=) The different types of distributed system are as follows:-
a) Distributed computing system
b) Distributed Information system
c) Distributed pervasive system

a) Distributed Computing System
An important class of distributed systems is the one used for high performance computing tasks. It is divided into two types.

i) Cluster Computing
ii) Grid Computing

Cluster Computing
It is a type of computing in which a group of computers are linked together so that they can act as single entity. In this computing the underlying hardware consists of a collection of similar workstations or PCs, closely connected by a means of a high speed local-area network. In addition, each node runs the same operating system.
For eg: the internet, search engine, Google uses cluster computing to provide reliable and efficient internet search services.

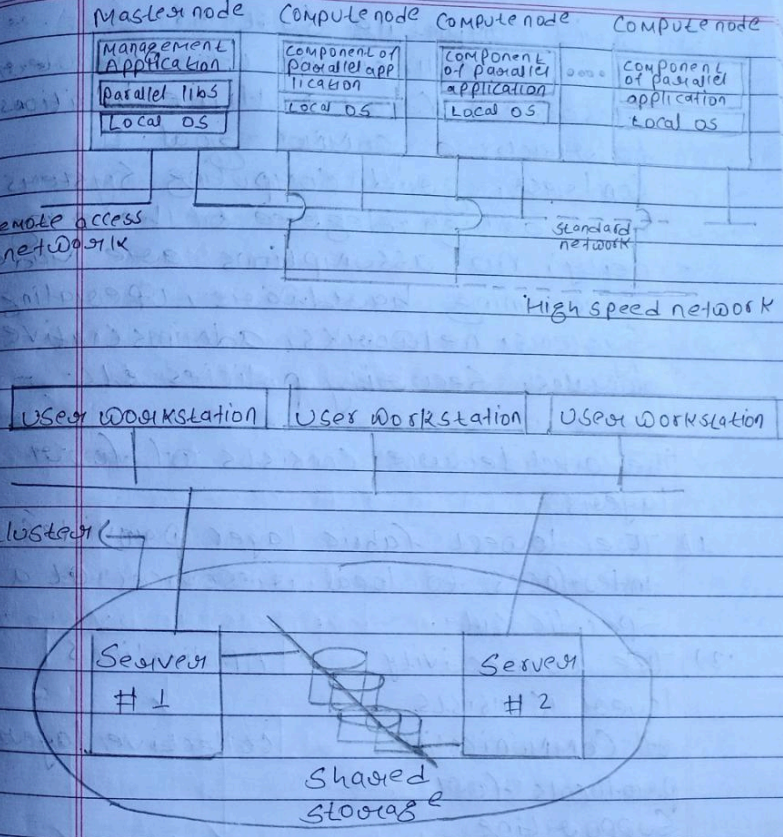

Fig: Cluster Computing

Grid Computing Systems

It is a collection of computer resources from multiple locations to reach a common goal. In contrast, grid computing systems have a high degree of heterogeneity; no assumptions are made concerning hardware, operating systems, networks, administrative domains, security policies etc.

The architecture consists of four layers:

1) The lowest fabric layer provides interfaces to local resources at a specific site.
2) The connectivity layer consists of communication protocols for supporting grid transactions that span the usage of multiple resources. For example, protocols are needed to transfer data between resources, or to simply access a resource from remote location.
3) The resource layer is responsible for managing a single resource. It uses the functions provided by the connectivity layer and calls directly and the interfaces made available by fabric layer.
4) The next layer in the hierarchy is the collective layer. It deals with handling access to multiple resources and typically consists of services for resource discovery, allocation and scheduling of tasks onto multiple resources, data replication, and so on.
5) Finally, the application layer consists of the applications that operate within a virtual organization and which make use of the grid computing environment.

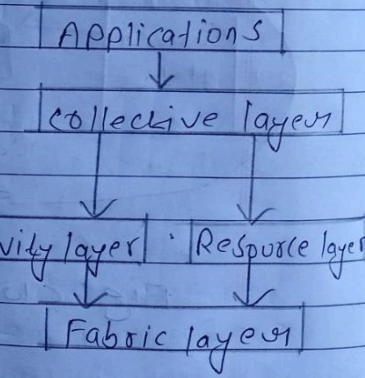

Fig: Showing layers

b) Distributed Information System
It is a computer service that runs at a single central location is more likely to become unavailable than a service ~~distributed~~ distributed to many sites. It can be explained further as:

*) Transaction processing system provides a highly structured client-server approach for database applications. More specifically, transactions are:

(i) Atomic
To the outside world, the transaction happens indivisibly.

(ii) Consistent
The transaction does not violate system invariants.

(iii) Isolated
Concurrent transactions do not interfere with each other.

(iv) Durable
Once a transaction commits, the changes are permanent.

Nested transaction is a transaction within another transaction (a sub transaction). For example: a transaction may ask for two things (airlines reservation info + hotel info) which would spawn two nested transactions.

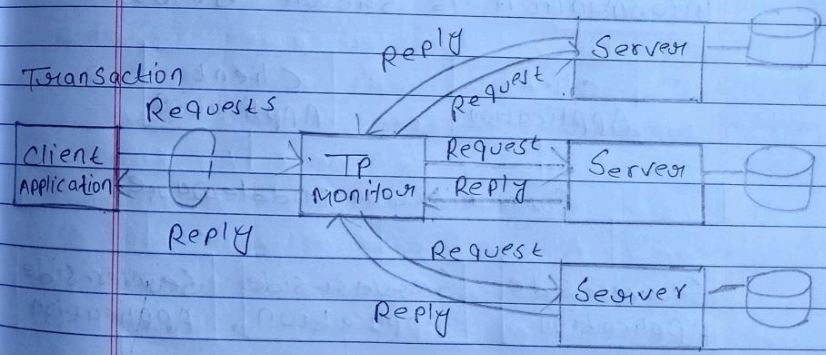

Fig: Transaction processing system

*) Enterprise Application Integration

In particular, application components should be able to communicate directly with each other and not merely by means of the request/reply behaviour that was supported by transaction processing systems.

The main idea that existing applications could directly exchange information is shown below.

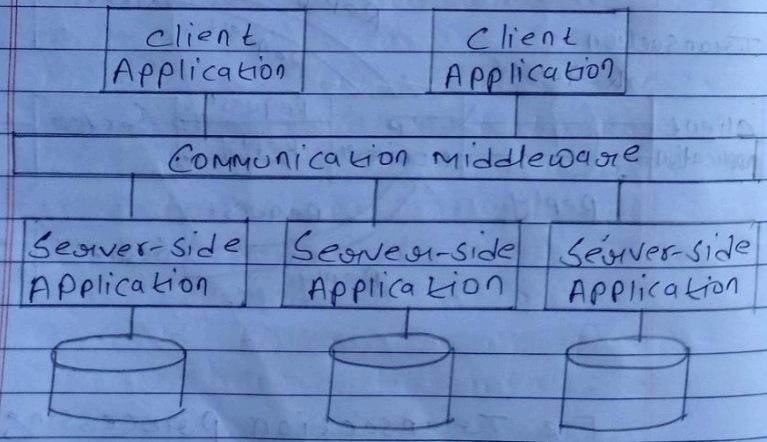

Fig: Enterprise Application Integration

Middleware in the context of distributed applications is software that provides services beyond those provided by the operating system to enable the various components of a distributed system to communicate and manage data. Middleware includes Web servers, application servers, messaging and similar tools that support application development and delivery.

c) Distributed pervasive system

The devices in these, what we refer to as distributed pervasive systems, are often characterized by being small, battery powered, mobile and having only a wireless connection, although not all these characteristics apply to all services. A distributed system is part of our surroundings.

Some concrete examples of pervasive system are as :-

1) Home Systems
2) Electronic Health Care Systems
3) Sensor Network

Home Systems

These systems generally consist of one or more personal computers, but more importantly integrate typical consumer electronics such as TVs, audio and video equipment.

Electronic Health Care Systems

With the increasing cost of medical treatment, new devices are being developed to monitor the well-being of individuals and to automatically contact physicians when needed. In many of these systems, a major goal is to prevent people from being hospitalized

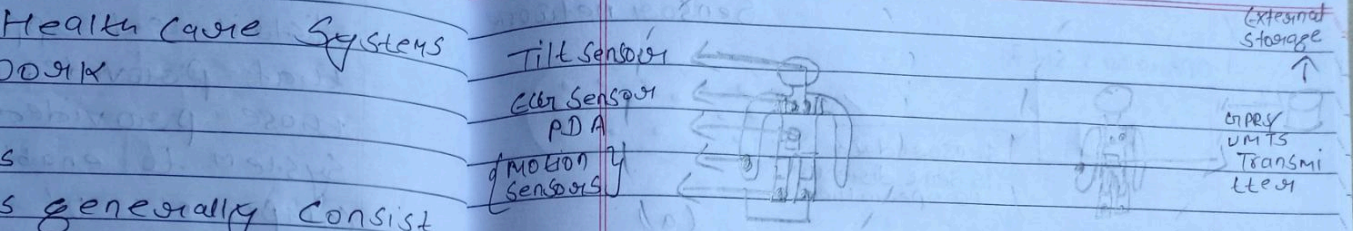

Fig: Electronic health care system

Sensor Networks

These networks in many cases form part of the enabling technology for pervasiveness and we see that many solutions for sensor networks return in pervasive applications. What makes sensor networks interesting from a distributed system's perspective is that in virtually all cases they are used for processing information.

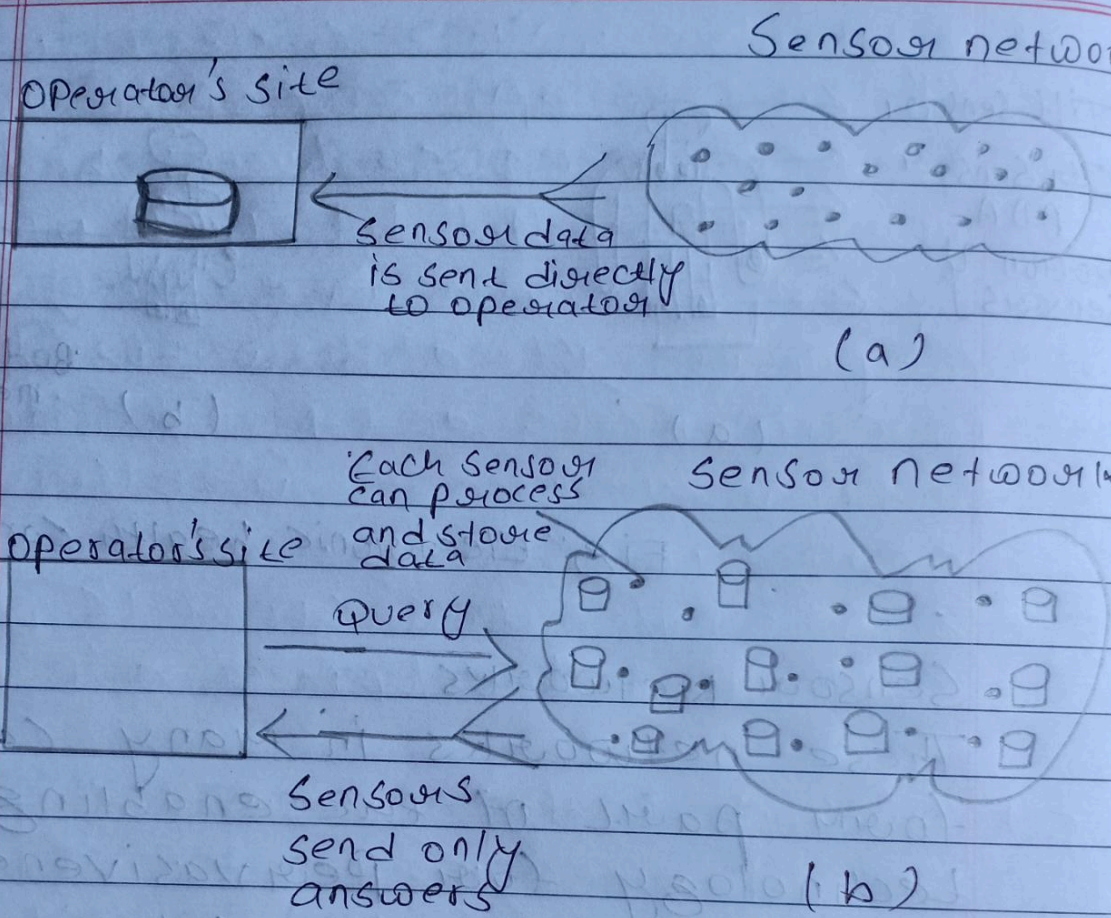

Fig: Showing example of how sensor network work.