

Parallel & Distributed Computing

Chapter # 1

Introduction.

⇒ Overwhelming pace of continuous change

- increasingly powerful computers
- high speed networks.
- smaller size

⇒ Increasing digitization of society

- how many computers are being used?
- mobile computers?

⇒ Distributed vs Decentralized systems

Types of networked Computer Systems

i) Decentralized Systems

- processes and resources are necessarily spread across multiple computers.

ii) Distributed Systems

- Processes and resources are sufficiently spread across multiple computers.

\Rightarrow Decentralization

- spreading implementation across multiple computers.
- is not end goal
- implementation decision that needs to be considered carefully.
- focus on sufficiently for spreading processes and resources.
- less spreading is generally preferred.

Examples of Decentralized Systems

1) Geographically Dispersed Systems

- monitoring physical locations like power plants.
- decentralization arises from spatial necessity of spreading processes and resources.

2) Distributed Ledger

Blockchain
1

- lack of mutual trust b/w participants
- uses publically verifiable transaction mechanism

- decentralization arises due to trust issues.

Distributed Systems Example

1) Google Mail Services

- access via web interface or mail clients
- incoming and outgoing server
imap.gmail.com
smtp.gmail.com
- estimated 10,000 mails per second are processed.

2) Network Attached Storage (NAS)

- smaller distributed system
- typically 2-4 hard drives.
- functions as file server.

Challenges

- complexity
- Partial failures
- security

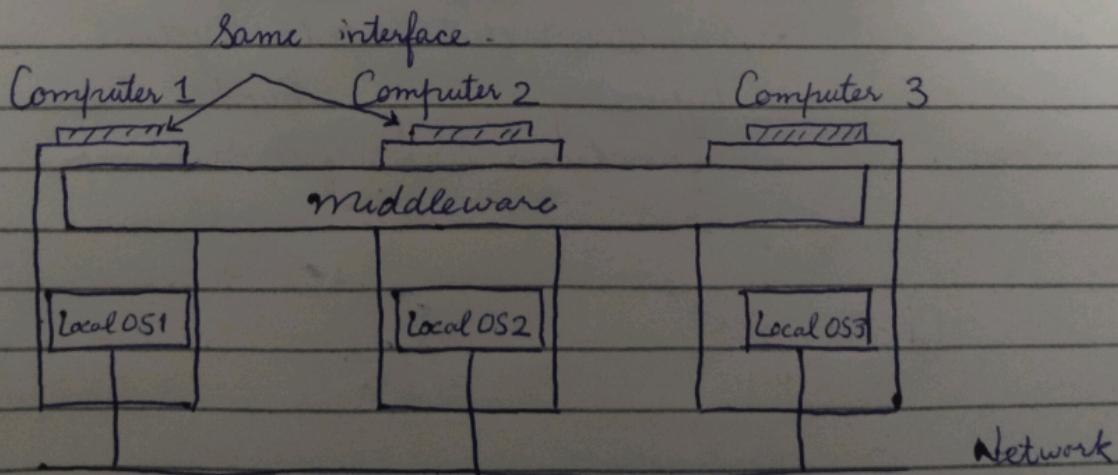
Design Goals of Distributed Systems

1) Resource Sharing

- make remote resources easily accessible
- economic benefits
- collaborative work.

2) Distribution Transparency

- hiding complexity of resource distribution
- resources are invisible to end users and applications.
- achieved through middleware maintaining consistent interfaces while hiding the implementation details.



Transparency

Description

Access

hide diff. in data representation and how it is accessed.

Location

hide where object is located

Relocation

hide that object may be moved to another location while in use.

Migration

hide that object may move to another location.

Replication

hide that object is replicated

Concurrency

hide that object may be shared by several users

Failure

hide failure & recovery of object.

Degree of Transparency

- ⇒ Aiming at full transparency can be too much.
 - comm. latencies cannot be hidden
 - completely hiding failures is impossible
- ⇒ Full transparency will cost performance
 - keeping replicas upto-date will consume time.
 - flushing of write operations
- ⇒ Exposing Distributive may be good
 - location based services.

3) Openness

- system components are easily usable and integrable.
- Standards of interoperability using standard rules, syntax, and semantics using Interface Definition Language (IDL).

4) Dependability

- Fault tolerance
- masks failures and ensure seamless recovery
- ⇒ fault classification
 - temporary
 - recovering

- permanent

Real work examples - harddrive.

Terms

Fault - cause of an error

Error - component that leads to failure

Failure - not working upto the specifications.

Fault Prevention

Fault Tolerance

Fault Removal

→ Fault Forecast

Metrices

• MTTF

• MTTR

(mean time to repair)



(mean time to failure)

Mean Time between failures

$$MTBF = MTTF + MTTR$$

5)

Security

- Crucial for dependability

- Authentication

- verify identity

- Authorization

- determine right to access certain resources.

- Cryptography
 - key technique for securing data.
- Secure Channels
 - HTTPS.

Chapter no 1

Design Goals

6. Scalability

- Handling growth in users, resources, and geography such that system maintains its performance.

i) Size Scalability

- ability to add more users without degrading performance.

ii) Geographical Scalability

- maintains performance despite significant communication delays due to distances.

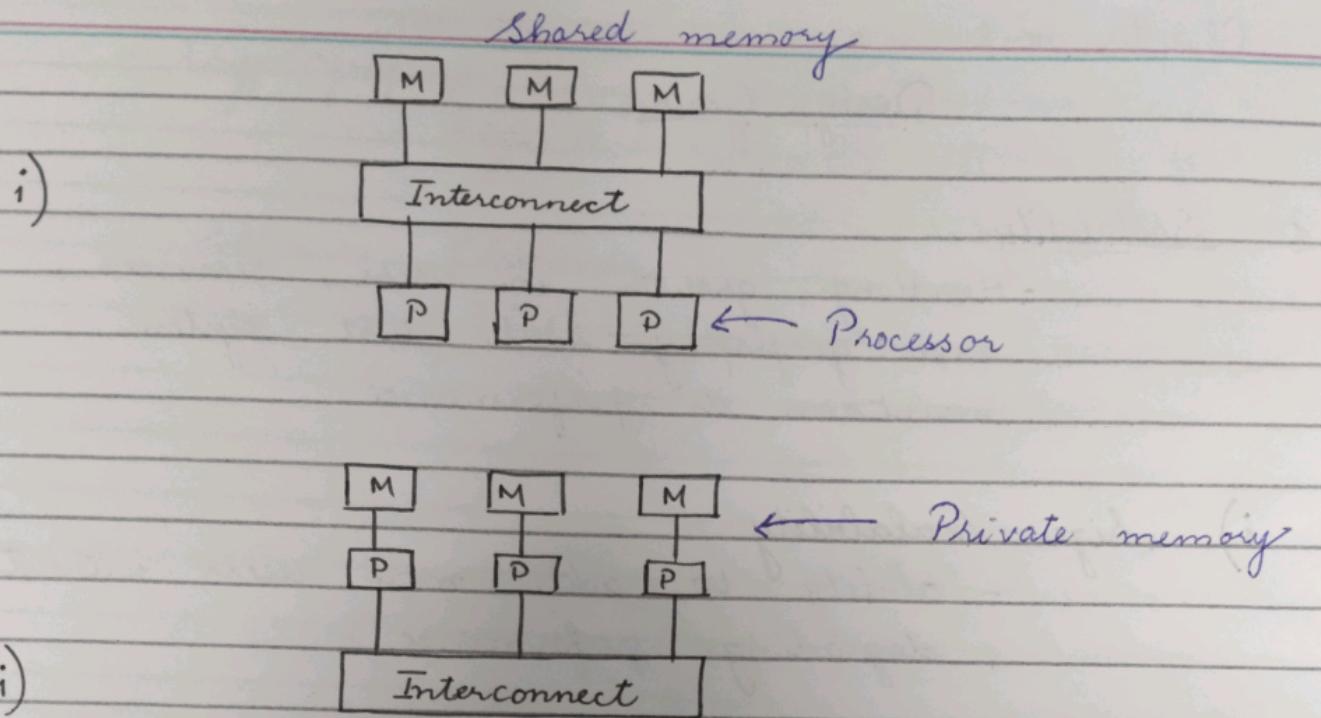
iii) Administrative Scalability

- easy management across multiple independent administrative domains

Classification of Distributed Systems.

1) High Performance Distributed Computing

- High Performance computing (HPC)
- Involves Parallel Processing
 - i)- Multiprocessor machines.
 - ii)- Multicomputer systems.



⇒ Cluster Computing

- group of high-end systems connected through high-speed interconnect (LAN).
- homogeneous → same OS & hardware
- tightly coupled managing nodes.

Examples

- weather forecasting
- Computation Fluid Dynamics.
- Gene sequencing
- Astro physics.

El-Capitan

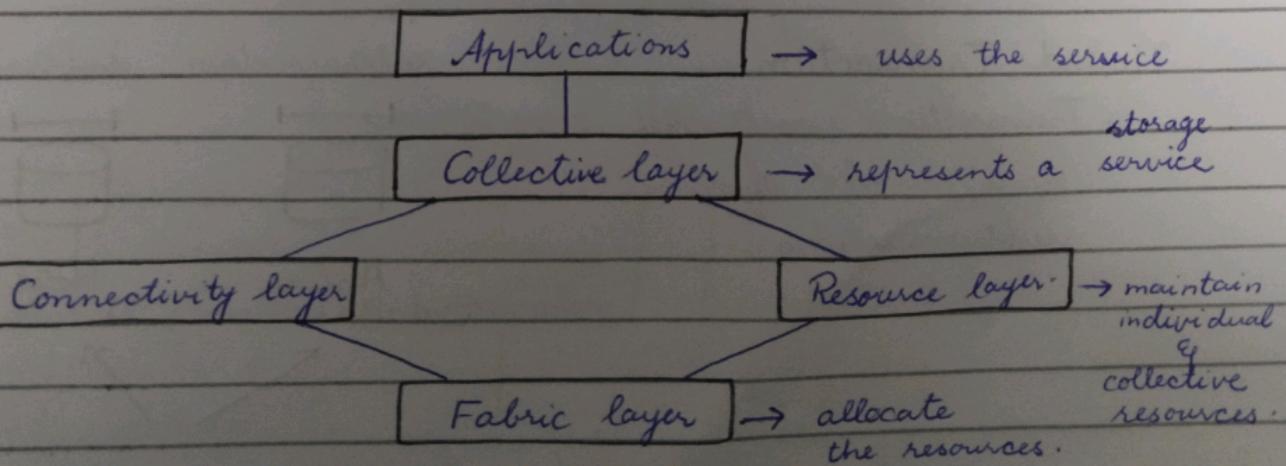
top 500. 10 m.

- CA USA 2024
- \$600 million USD
- 43808 AMD EPIC 24C @ 1.8 GHz
- Total memory of 5.43 PB
- 2746 PFLOPs
- 30 MW

⇒ Grid Computing.

- Heterogeneous federation of systems
- Dispersed across several organizations.
- Spans over a wide area network (WAN)

layered Architecture of Grid Computing.



Chapter no. 1

Classification of Distributed Systems

1)
2)

HPDC

Distributed Information Systems

- Integrate applications into an enterprise-wide information system using middleware solutions.

- Distributed Transaction Processing
→ combining multiple requests into a single distributed transaction

→ ACID

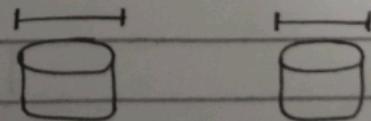
↳ atomic, consistent, Isolated, durable.

→ all or nothing.

Nested Transaction

- Transaction constructed as a number of subtransactions.

Nested Transaction
+-----+
| subtransaction | subtransaction |



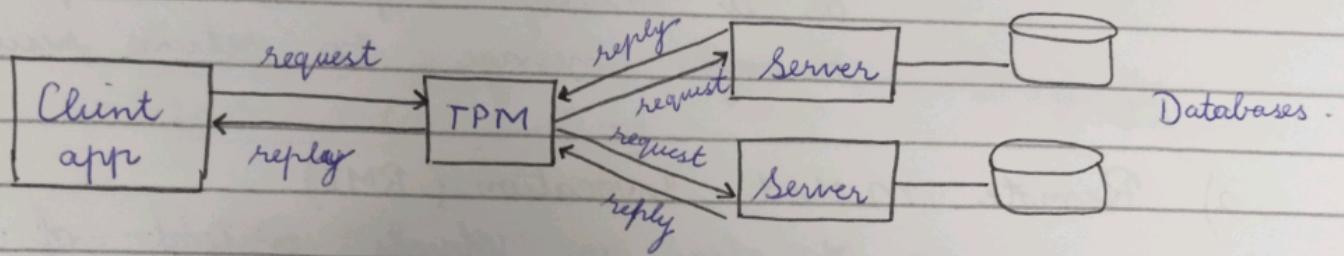
Airline DataBase Hotel DataBase

← →

Two different independent databases.

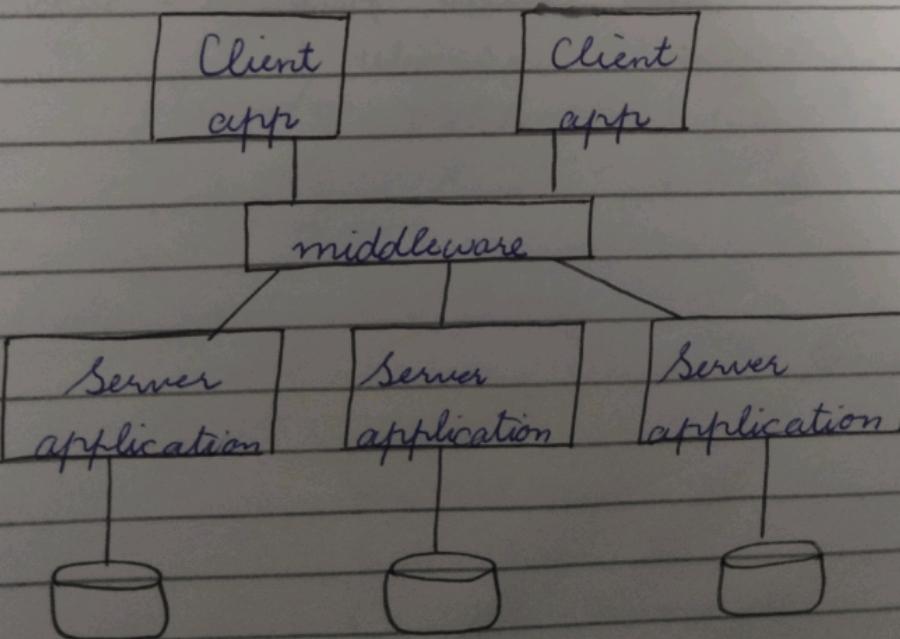
Transaction Processing Monitor (T.P.M)

- coordinates the execution of a distributed transaction across several servers.



Enterprise Application Integration (EAI)

- direct communication between independent components is essential leading towards EAI
- middleware facilitates communication.



Types of communication middleware

1) Remote Procedure Call (RPC)

- requests are sent through a local procedure call, packaged as a message, processed, responded through message to return results.

2) Remote Method Invocation (RMI)

- operates on objects instead of functions.

3) Message Oriented Middleware (MOM)

- Messages are sent to logical contact points (published) and forwarded to subscribed applications.
- Also called publish-subscribe systems.

Classification of Distributed Systems.

- 1) HPDC
- 2) DIS
- 3) Pervasive Systems

- Evolution from Traditional to Emerging Distributed Systems

- HPDC & DIS

- fixed nodes
- high quality network.

- Pervasive Systems

- mobility
- embedded into users environment

Types

- 1) Ubiquitous computing
- 2) Mobile computing
- 3) Sensor network.

1) Ubiquitous Computing

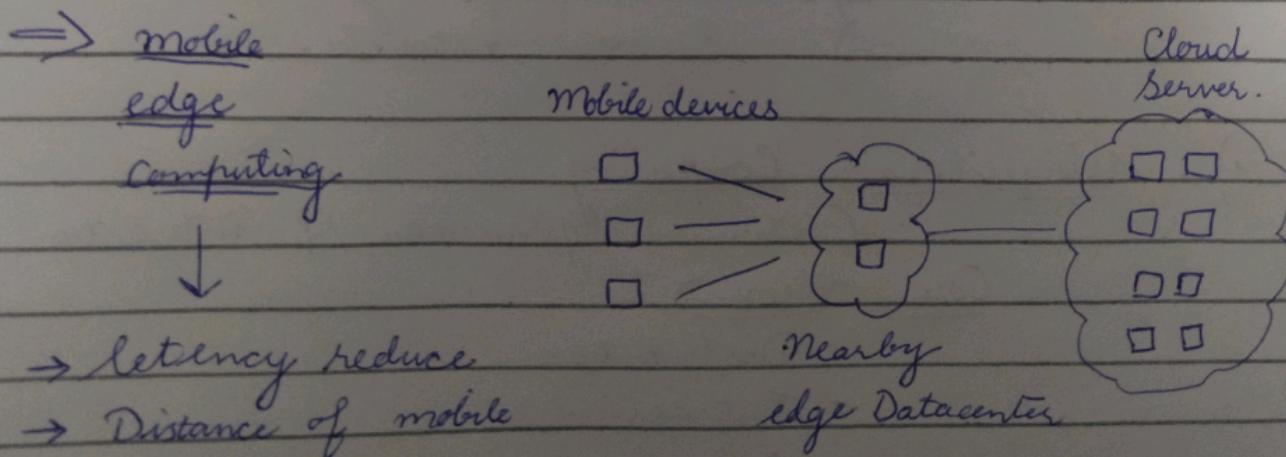
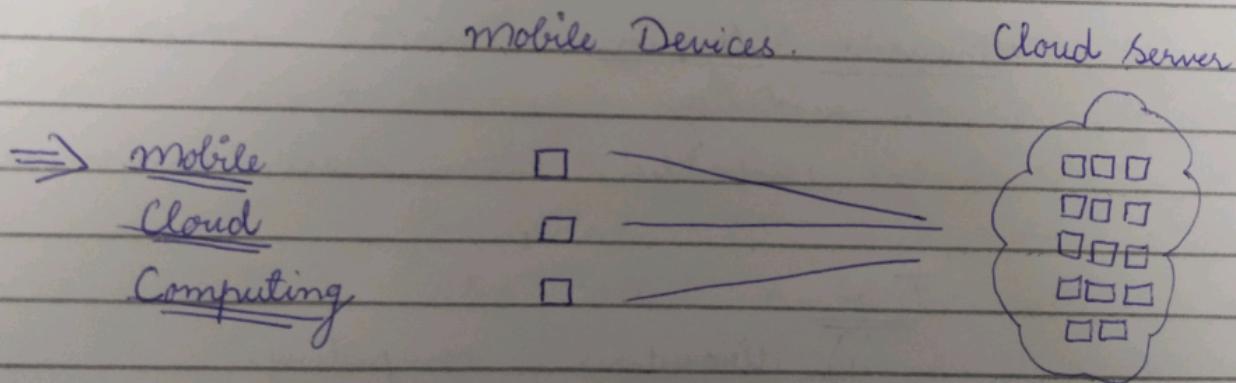
- Continuous presence of interaction.

Core Elements

- Distribution
- Interaction
- Content awareness
- Autonomy
- Intelligence.

Mobile Computing

- Diverse set of devices.
- Dynamic location & server discovery.
- Changing location.
- Reachability and server discovery.
- location awareness.
- Resource and Service sharing among mobile devices
 - not widely adopted.

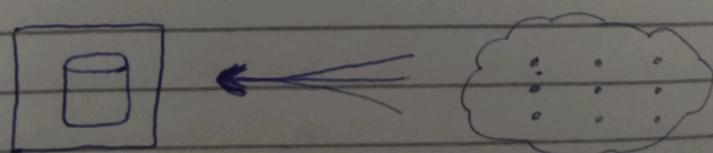


Sensor Networks

- sensor nodes collaborate to process sensed data efficiently in an application specific manner.

Characteristics

- Many nodes equipped with one or more sensors.
- can act as actuators.
- often battery powered.
- use wireless communication.
- simple device with limited resources.
- Sensor networks organized as distributed databases.
- Two extremes:
 - i) Storing & processing data at operators site.
 - ii) Storing & processing data at sensors.

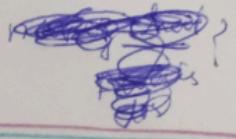


Operator's site

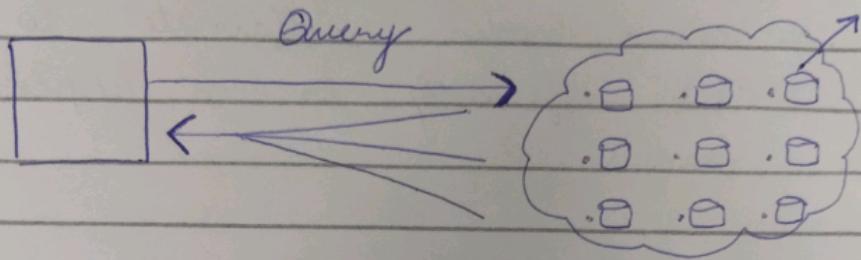
Sensor Network

Use \Rightarrow forward data continuously.

\rightarrow Resource wastage more.



operator site

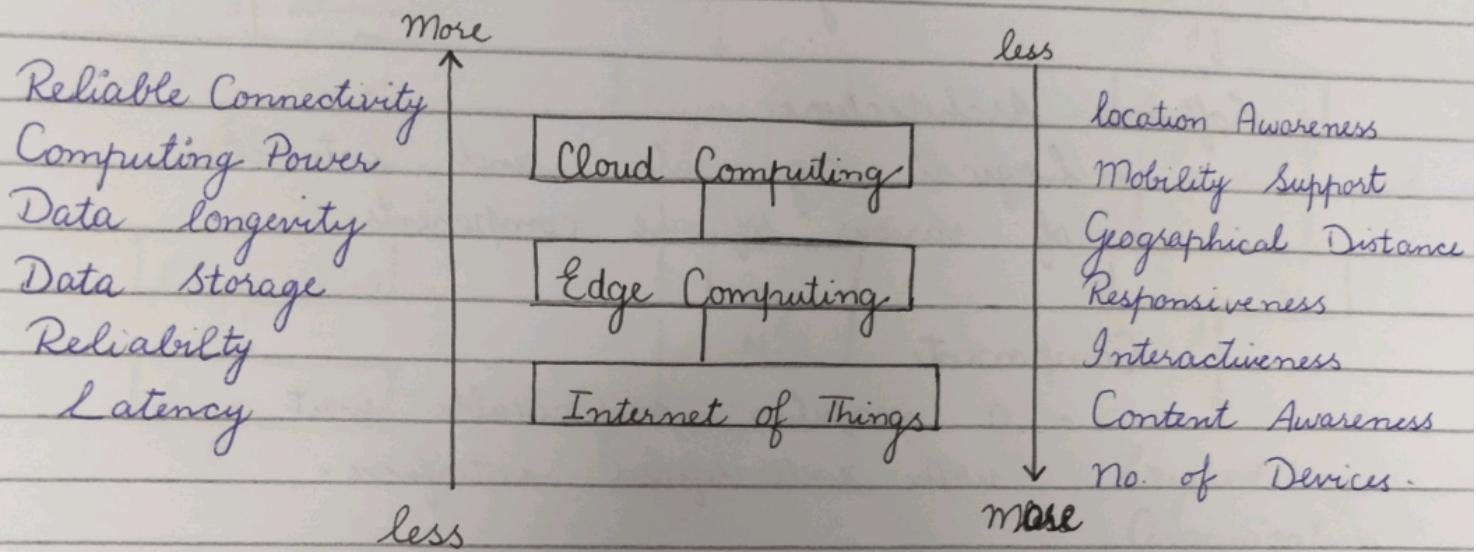


Each sensor
can process
and store
data.

⇒ When we send query then sensors send data to operator site but in first approach sensors send data to operator site whether we ask or not.

Chapter no. 1

Hierarchical view from clouds to devices.



Developing Distribute systems : Pitfalls.

- False assumptions often made when developing a distributed application.
 - network is reliable
 - network is secure
 - network is homogeneous.
- Topology does not change
- Latency is zero.
- Bandwidth is infinite
- Transport cost is cost.
- There is one administrator.