Lecture 01 · SENG 41283 · Distributed and Cloud Computing

Characterization of Distributed Systems

Definition · Three Core Characteristics · Desired Properties · Types · Trends · Challenges · Transparency

1. Introduction and Definition

A distributed system is one in which components located at networked computers communicate and coordinate their actions only by passing messages. Computers connected by a network may be spatially separated by any distance. Three especially significant characteristics follow from this definition.

1.1 Concurrency

In a network of computers, concurrent program execution is the norm. Resources such as web pages or files are shared when necessary; capacity can be increased by adding more computers to the network. Coordinating concurrently executing programs that share resources is a critical design challenge.

1.2 No Global Clock

Programs coordinate by exchanging messages. Close coordination depends on a shared idea of time, but there are fundamental limits to the accuracy with which computers in a network can synchronise their clocks. There is NO single global notion of the correct time — a direct consequence of communication only by message passing.

1.3 Independent Failures

Faults in the network isolate computers, but those computers do not stop running. Programs on isolated computers may not detect whether the network has failed or has become unusually slow. The failure of a computer or process crash is not immediately known to other components. Each component can fail independently, leaving the others still running.

2. Desired Properties

Property	Description
Fault-Tolerant	Recovers from component failures without performing incorrect actions.
Highly Available	Restores operations and resumes services even when some components have failed.
Recoverable	Failed components can restart and rejoin the system after the cause of failure is repaired.
Consistent	Coordinates actions by multiple components in the presence of concurrency and failure.

Scalable	Operates correctly as the system is scaled to a larger size; load increases should not have a significant effect.
Predictable Performance	Provides the desired responsiveness in a timely manner.
Secure	Authenticates access to data and services.
Quality of Service	Provides guarantees on qualities such as performance, security, and reliability.

3. Types and Examples

- **Client–Server:** clients contact the server for data; changes committed back to server.
- **Three-Tier:** information stored in a middle tier; most common for web applications.
- **N-Tier:** used when an application needs to forward requests to additional enterprise services.
- **Peer-to-Peer:** responsibilities uniformly distributed among peers; each peer acts as client or server.
- **Examples:** World-Wide-Web · Web Search · Massive Multiplayer Games · Financial Trading

4. Trends in Distributed Systems

Pervasive Networking

The Internet is a very large distributed system consisting of intranets protected by firewalls and linked by high-capacity backbone networks (satellite, fibre optic). ISPs provide broadband links to individual users and small organisations.

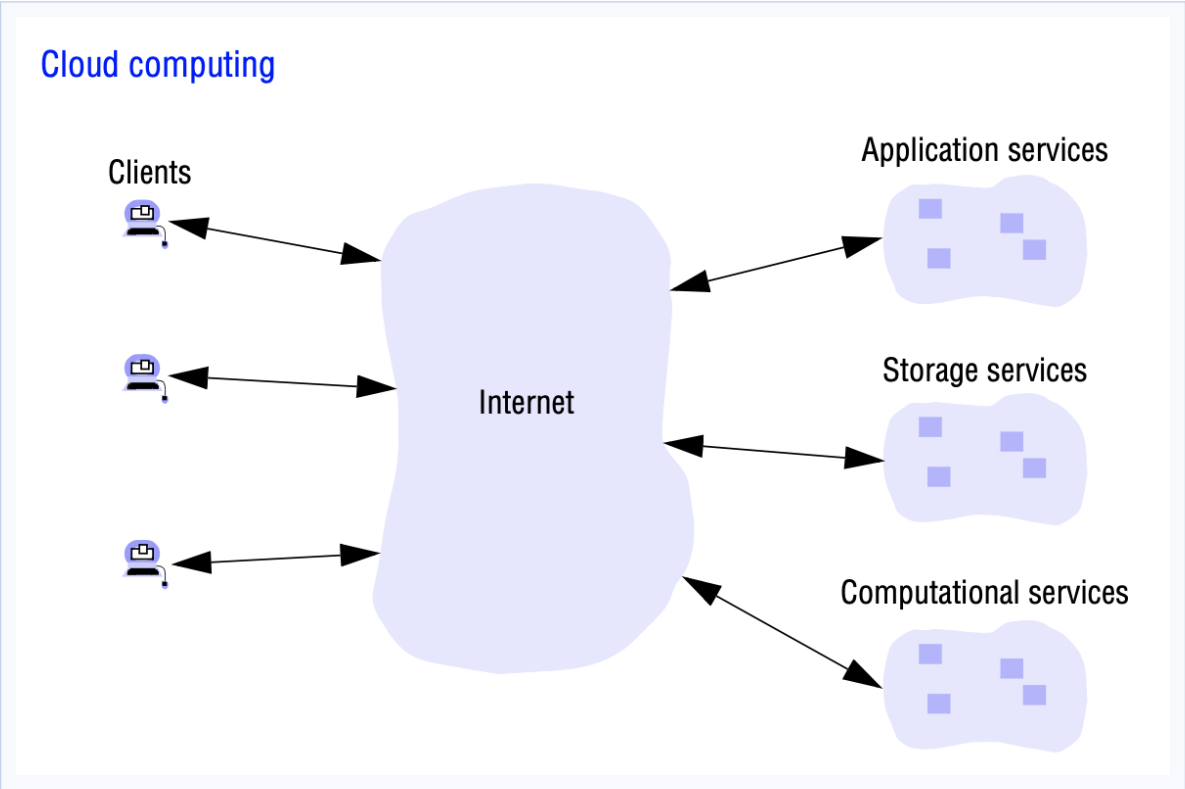

Figure 4.1 Internet topology: intranets linked by backbones and connected via ISPs.

Ubiquitous and Mobile Computing

Small, cheap computational devices are embedded in everyday physical environments. Mobile computing allows users away from their home intranet to access resources via devices they carry. Location-aware and context-aware computing enables use of nearby resources (printers, point-of-sale terminals).

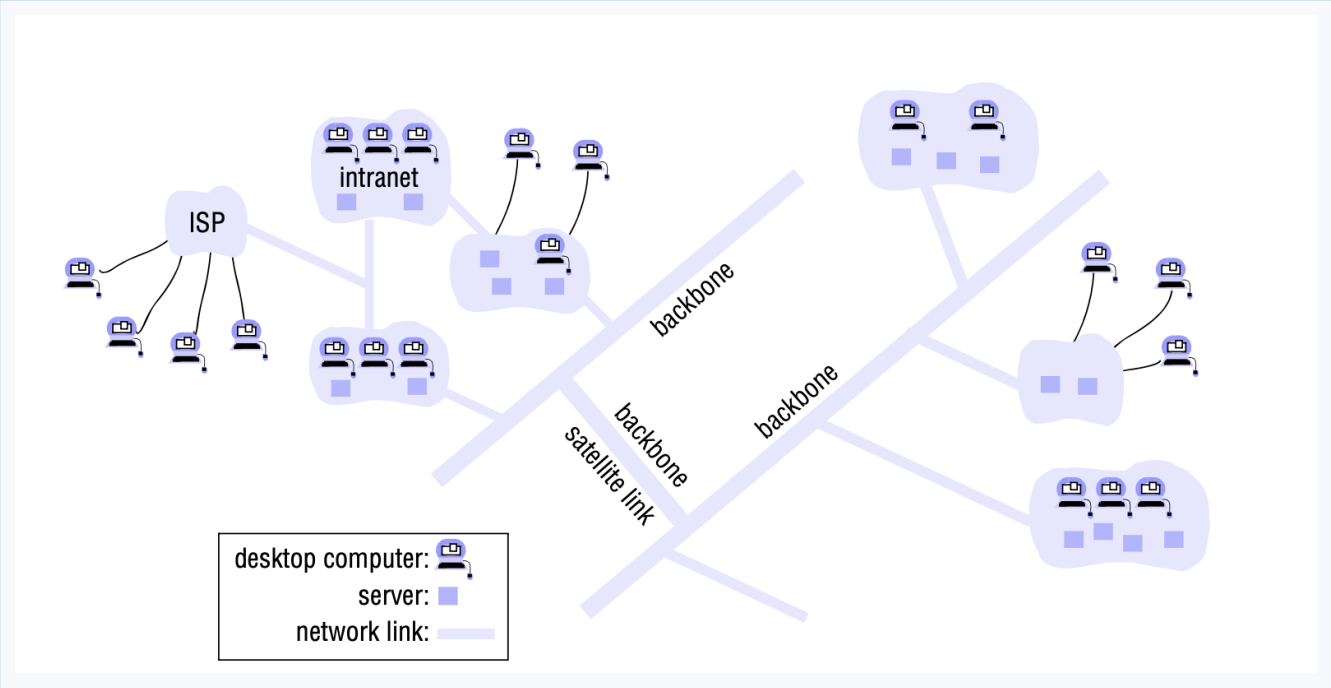

Figure 4.2 Ubiquitous and mobile computing: devices carried by the user connect to nearby infrastructure.

Multimedia Services

Distributed multimedia systems support both discrete media types (pictures, text) and continuous media types (audio, video). The temporal dimension is critical: video requires a given frame rate and maximum delivery latency. Services include video-on-demand, audio/video conferencing, IP telephony, and webcasting.

Distributed Systems as a Utility

Physical resources such as storage and processing are made available to networked computers on a rental basis (Amazon, Google). OS virtualisation is the key enabling technology. Software services are also available as utilities (Google Workspace).

5. Challenges

Heterogeneity

Must accommodate different networks, operating systems, hardware, and programming languages. Internet protocols mask network differences; middleware deals with the rest.

Openness

Systems should be extensible. Interfaces are published as RFCs (since the early 1980s). Open distributed systems can be extended at hardware and software levels.

Security

Confidentiality (protection against disclosure), **integrity** (protection against alteration), and **availability** (protection against denial of access). Persistent challenges: Denial-of-Service attacks; security of mobile code.

Scalability

A system is scalable if the cost of adding a user is a constant amount in resources. Challenges: controlling physical resource cost; avoiding performance bottlenecks (DNS uses hierarchical structure rather than a flat master file); preventing resource exhaustion (IPv4 shortage); decentralised algorithms.

Failure Handling

- **Detecting:** checksums; timeouts.
- **Masking:** retransmit lost messages; RAID disk pairs.
- **Tolerating:** inform the user; allow retry.
- **Recovery:** roll back permanent data to a consistent state.
- **Redundancy:** at least two routes between routers; DNS tables replicated in at least two servers.

Concurrency

Multiple concurrent users create races on shared resources. Each resource must be designed to be safe in a concurrent environment. Example: two concurrent auction bids interleaved without control may be stored with the wrong values.

Transparency

The aim is to make distribution invisible to the application programmer.

Type	Meaning
Access	Local and remote resources accessed using identical operations.
Location	Resources accessed without knowledge of their physical location.
Concurrency	Several processes operate concurrently using shared resources without interference.
Replication	Multiple instances used to increase reliability and performance.
Failure	Faults concealed, allowing tasks to complete despite hardware/software failures.
Mobility	Resources and clients can move without affecting the operation of users or programs.
Performance	System reconfigured to improve performance as loads vary.
Scaling	System and applications expand in scale without change to structure or algorithms.

Exercise

Write a blog post on a technology selected according to the last digit of your index number, covering: introduction, use of the technology, type of distributed system, and how it preserves the seven properties above.

Index	Topic
[0]	Blockchain
[1]	Hadoop

[2]	Redis Clustering
[3]	Content Delivery Network
[4]	HBase Database
[5]	Apache Spark
[6]	Kafka
[7]	Domain Name System
[8]	Google Search Engine
[9]	Torrent