1. Pitfalls, False and Hidden Assumptions in Distributed Systems

Distributed systems often rely on several assumptions that may not hold in real-world deployments:
- Reliable Network: Developers often assume the network will never drop or delay packets, but in reality, networks are prone to failures and congestion.
- Zero Latency: There is a hidden assumption that communication between nodes is instantaneous, which neglects real-world delays.
- Infinite Bandwidth: Assumes no limitation on data transmission speed, which is unrealistic especially in large-scale systems.
- Always Available Nodes: Assumes that all nodes are always up, failing to consider system crashes or power failures.
- Global Clock: Distributed systems often presume perfectly synchronized clocks, which is practically impossible without introducing clock drift.
- Identical Performance: Assumes all nodes perform equally well, ignoring hardware and network discrepancies.
- Security: Assumes internal network communication is safe, potentially overlooking malicious attacks.

2. Distributed vs Decentralized Systems

- Distributed System: Comprises multiple independent components located on different machines that communicate and coordinate their actions to appear as a single coherent system. A central authority may still exist.
  Example: Google Cloud

- Decentralized System: A subset of distributed systems where control is distributed; no single entity has full control.
  Example: Blockchain, BitTorrent

| Feature | Distributed System | Decentralized System |
|---|---|---|
| Control | May have central control | No central authority |
| Coordination | Coordinated actions | Autonomous peer actions |
| Failure Impact | Partial or complete | Localized |

3. Mobile Cloud Computing vs Mobile Edge Computing

- Mobile Cloud Computing (MCC) involves offloading data processing and storage to centralized cloud servers. While it supports scalability and flexibility, it often suffers from high latency and dependence on connectivity.

- Mobile Edge Computing (MEC) brings computation and storage closer to the user, often at the edge of the network (e.g., base stations). This improves responsiveness and supports real-time applications.

| Feature | Mobile Cloud | Mobile Edge |
|---|---|---|
| Latency | Higher | Lower |
| Processing Site | Remote cloud servers | Nearby edge devices |
| Use Cases | Cloud backup, Email | AR/VR, Self-driving cars |

4. Grid Computing

Grid computing is a form of distributed computing where resources from multiple domains are pooled together to reach a common objective. It allows tasks to be broken down and processed in parallel across multiple machines.
- Example: SETI@home
- Key Characteristics: Resource sharing, distributed nodes, heterogeneous systems.

5. Cluster Computing

Cluster computing involves a group of tightly coupled computers that work together to perform tasks as a single system. These systems are usually located close together and connected via high-speed networks.
- Example: Hadoop clusters, Beowulf clusters
- Key Characteristics: High availability, load balancing, parallel processing

6. Parallel Computing

Parallel computing is the simultaneous use of multiple compute resources to solve a computational problem:
- Types: Data Parallelism, Task Parallelism
- Goal: Increase speed and efficiency
- Example: Multi-core processors handling matrix multiplication

7. Policies vs Mechanisms

- Policies are the "what"  rules that govern system behavior (e.g., which process to run next).
- Mechanisms are the "how"  the implementation that enforces policies (e.g., context switching).
- Example: Scheduling policy (round robin) vs scheduling mechanism (timer interrupt).

8. Scale in Distributed Systems

Scalability refers to the system's ability to handle growth:
- Size Scalability: Adding more nodes
- Geographical Scalability: Nodes spread across locations
- Administrative Scalability: Support multiple organizations
- Example: Cassandra's ring topology supports horizontal scaling.

9. Ubiquitous Systems

Also known as pervasive computing, it aims to make computing available everywhere and anytime through embedded sensors and smart devices.
- Examples: Smart homes, wearable health monitors, IoT devices
- Features: Context-awareness, invisibility, mobility

10. Mobile Computing

Mobile computing allows users to access computational resources without a fixed location using portable devices.
- Components: Mobile devices, wireless networks, cloud services
- Examples: Smartphones, laptops with 4G/5G access

11. System Availability Calculation

Given subsystem availabilities: 60%, 70%, 80%, 90%
Availability in parallel:
A = 1 - (1 - A1)(1 - A2)(1 - A3)(1 - A4)
A = 1 - (0.4 * 0.3 * 0.2 * 0.1) = 1 - 0.0024 = 0.9976 = 99.76%

12. P2P Architectural Structures

- Pure P2P: All nodes are equal (BitTorrent)
- Hybrid P2P: Centralized indexing with P2P content sharing (Napster)

Diagrams:
- Pure P2P: Mesh network where every node connects to others
- Hybrid P2P: Clients connect to server for metadata, then peer-to-peer for files

## 13. Hybrid System Architectures

Combines client-server and P2P or edge-cloud structures:
- Example 1: Edge devices perform initial processing; cloud stores data
- Example 2: Spotify uses central server for discovery and P2P for streaming

## 14. Middleware Architectures

Middleware connects applications and services:
- Message-Oriented: Kafka, RabbitMQ
- Remote Procedure Call (RPC): gRPC, XML-RPC
- Object Middleware: CORBA, Java RMI
- Database Middleware: ODBC, JDBC

## 15. Project Relevance with Course Concepts

Your project on Cassandra touches many Distributed & Parallel concepts:
- Distributed Systems: Cassandras architecture spans multiple nodes
- Replication & Consistency: Testing eventual consistency models
- Scalability: Horizontal scaling of nodes
- CAP Theorem: Cassandra trades off consistency for availability (AP)
- Fault Tolerance: Node failure recovery via replication
- Parallel Querying: Multiple consistency-check operations run in parallel
- Middleware: Internal services coordinate read/write consistency across clusters