Consistency in CAP Theorem (NoSQL Systems)

In contrast, **the "C" in CAP theorem** applies to **ordering operations on a single data item**, ensuring that **all replicas eventually converge to the same state**. CAP theorem states that in a distributed system, you can achieve only two out of three properties:

1. **Consistency (C)** – Every read receives the most recent write or an error.
2. **Availability (A)** – Every request gets a response, even if some nodes are down.
3. **Partition Tolerance (P)** – The system continues to function even if network failures occur.

Since network failures are **inevitable**, distributed systems must choose between **strong consistency (C) and availability (A)**.

- **In traditional databases, strong consistency is enforced, sacrificing availability.**
- **In NoSQL and Big Data systems, availability is prioritized, leading to eventual consistency.**

The **consistency in CAP theorem is a strict subset of ACID consistency**. It ensures that **replicas of a single data item** eventually become identical, but it does not guarantee complex transactional consistency across multiple records.

Question 1: Stock Trading Platform (5 Marks)

Consider a stock trading platform used by financial institutions or brokers to execute trades and manage stock portfolios. This platform operates within a single data center or closely networked data centers that are not geographically dispersed. In this scenario, refer to CAP theorem to briefly justify which type of database configuration is preferred among AP, CP, and CA.

In a stock trading platform, **data correctness and precision are critical**. A small inconsistency, even by milliseconds, can lead to major financial losses or regulatory violations. The platform operates within a **single data center or tightly connected data centers**, which significantly **reduces the chance of network partition (P)**.

Given this controlled network environment, it becomes feasible to assume that **Partition Tolerance is not a hard requirement** in this case. This opens up the possibility of focusing on **Consistency (C)** and **Availability (A)** together — essentially aiming for a **CA configuration**.

In this CA configuration:

- **Consistency (C)** ensures that when a trade is executed or a portfolio is updated, **all users see the exact same data** at the same time.
- **Availability (A)** ensures that the system remains responsive — traders can always place or query orders.

Since the **risk of partitioning is minimal** in a localized setup, this trade-off is acceptable. Hence, **CA is the preferred choice** for a stock trading platform operating in a centralized environment.

Question 2: Ride-Hailing Platform (5 Marks)

Consider a ride-hailing platform (such as Uber or Ola) used to connect passengers with drivers. This platform operates across multiple cities and may encounter temporary network partitions during peak hours or in congested areas. In this scenario, refer to the CAP Theorem to briefly justify which type of database configuration is preferred among AP, CP, and CA.

A ride-hailing platform like Uber or Ola operates in a **distributed and highly dynamic environment**. It spans multiple geographic locations and frequently experiences **network latency and partitions**, especially during peak traffic hours or mobile data congestion.

In such a scenario, **Partition Tolerance (P)** becomes **non-negotiable**. The system must continue functioning even when network issues occur between data centers or edge nodes.

Between **Consistency (C)** and **Availability (A)**, the ride-booking system must prioritize **Availability**.

- If a user cannot book a ride during a peak hour due to temporary inconsistency in driver location data, that's **preferable** to not being able to use the service at all.
- Slight inconsistencies (like showing a nearby driver who just accepted another ride) can be resolved in subsequent refreshes or retries.
- The system must **remain responsive**, even under degraded connectivity, to maintain customer satisfaction and driver engagement.

Therefore, **AP (Availability + Partition Tolerance)** is the preferred configuration. The system **trades off strict consistency** in favor of continued availability and responsiveness, which is acceptable for this use case.

Question 3: Online Retail Platform (5 Marks)

Consider an e-commerce platform like Amazon that handles millions of user requests for product searches, inventory updates, and order placements across globally distributed data centers. During a major sale event, some services may experience network partitions. In this scenario, refer to the CAP Theorem to briefly justify which type of database configuration is preferred among AP, CP, and CA.

In this case, **Partition Tolerance (P)** is a must because the platform operates globally across distributed data centers, where network partitions can occur, especially during high-traffic sale events.

Between **Consistency (C)** and **Availability (A)**, **Availability** is typically prioritized. For example, a slightly stale inventory count is tolerable if it allows users to continue browsing and placing orders. Immediate feedback is more important than strict real-time accuracy in this context.

Thus, an **AP configuration** is preferred. The system tolerates partitions and maintains service availability, even if some data inconsistencies exist briefly — these can be resolved asynchronously.

Question 4: Social Media Feed Service (5 Marks)

A social networking platform allows users to post content and view real-time updates from friends. The system is globally distributed and is expected to remain accessible even during regional outages. In this context, apply the CAP Theorem to justify whether AP, CP, or CA configuration is most suitable for maintaining user experience.

Social media platforms prioritize **user engagement and responsiveness**, especially under high loads or regional failures. Given its globally distributed nature, **Partition Tolerance (P)** is required.

Between **Availability (A)** and **Consistency (C)**, social platforms often prioritize **Availability**. Minor inconsistencies (e.g., seeing a post a few seconds later or in a slightly different order) are acceptable trade-offs if the system remains usable.

Therefore, an **AP configuration** is most suitable. It allows the platform to serve users continuously during network partitions, even if some updates are temporarily inconsistent across regions.

Question 5: Banking Transaction System (5 Marks)

A core banking application processes account transfers, ATM withdrawals, and real-time balance updates. The system runs in a secure, tightly coupled infrastructure with minimal tolerance for data loss. Based on the CAP Theorem, justify which configuration — AP, CP, or CA — is most suitable for such a system.

In banking systems, **Consistency (C)** is non-negotiable. Any inconsistency in transaction history or balance updates could result in financial loss or compliance violations.

Since the system is hosted in a **tightly coupled, localized infrastructure**, **Partition Tolerance (P)** can be minimized or assumed less critical.

Therefore, **CA (Consistency + Availability)** is a valid configuration. It ensures that data is always correct and accessible, given that the system can operate with high network reliability.

Question 6: Real-Time Multiplayer Game Server

A real-time multiplayer game server supports thousands of players interacting live across global regions. The game must handle temporary network slowdowns without significantly disrupting gameplay. Use the CAP Theorem to justify the preferred configuration (AP, CP, or CA) for such an application.

A game server must maintain **Availability (A)** to keep players connected and responsive. Because global network slowdowns and partitions are likely, **Partition Tolerance (P)** is also essential.

However, **strict consistency (C)** is less important in real-time games, as players can tolerate occasional state mismatches or "lag" corrections during gameplay.

Hence, an **AP configuration** is ideal. It prioritizes availability and partition tolerance, ensuring uninterrupted user experience even with some temporary inconsistencies in game state.

Question 7: Health Monitoring System for ICU Patients

A hospital's ICU monitoring system collects and updates real-time patient vitals. Doctors rely on this system for immediate intervention. The system is hosted within the hospital's local network. Using the CAP Theorem, justify which configuration — AP, CP, or CA — should be prioritized for this use case.

In an ICU monitoring system, **Consistency (C)** is critical — any delay or error in patient vitals could jeopardize lives. The system must always display the most up-to-date and accurate readings.

Because it operates within a **local, highly reliable network**, **Partition Tolerance (P)** can be de-emphasized.

Therefore, a **CA configuration** is suitable. The system delivers consistent and available data within the hospital's network, ensuring both accuracy and responsiveness for medical professionals.

Question 8: Collaborative Document Editing Platform

A cloud-based collaborative document editing system allows multiple users to simultaneously edit and view documents in real-time from different locations. Apply the CAP Theorem to justify which configuration — AP, CP, or CA — is best suited for such a system.

Real-time collaborative systems operate in a **distributed environment** with a need for high responsiveness. Therefore, **Partition Tolerance (P)** is necessary.

While availability is important, **strong consistency is more critical** to avoid conflicting edits. For example, users editing the same paragraph should not overwrite each other's work unpredictably.

Thus, a **CP configuration** is preferred. The system ensures consistent document state across users, even if that means temporarily blocking access during partition events to avoid conflicting updates.

Question 9: DNS (Domain Name System)

DNS is a globally distributed system that resolves domain names into IP addresses. Use CAP Theorem to determine which configuration it follows and why.

The DNS system must remain **highly available** — users around the world must be able to resolve domains even if some servers are unreachable. Since it's a distributed system, **Partition Tolerance (P)** is also essential.

DNS is eventually consistent: it may return outdated records during cache propagation, but this is acceptable for most users.

Thus, DNS follows an **AP configuration** — prioritizing **Availability** and **Partition Tolerance**, while sacrificing strong consistency in favor of eventual consistency.

Question 10: Distributed Logging System

A company uses a distributed logging system to collect application logs from thousands of microservices in real-time. Delayed delivery is acceptable, but loss of logs or duplication is not. Based on CAP Theorem, which configuration — AP, CP, or CA — should this system use and why?

In a logging system, **Consistency (C)** is important to ensure that all logs are captured correctly and not lost or duplicated. Because the system operates across distributed services, **Partition Tolerance (P)** is required.

Availability can be temporarily sacrificed — it's acceptable if some nodes are temporarily unable to query logs as long as they are stored correctly.

Hence, a **CP configuration** is most appropriate. It ensures log integrity even during network partitions, aligning with operational and audit requirements.

Question 11: Distributed File Storage (e.g., HDFS or Google File System)

A distributed file system stores large-scale data for batch processing. Occasional delays are tolerable, but file consistency must be guaranteed across nodes. Apply CAP Theorem to identify the most suitable configuration and justify.

In distributed file systems, **Partition Tolerance (P)** is essential due to node failures and network variability. **Consistency (C)** is also crucial — data must not be corrupted or diverge across replicas.

Availability can be traded off briefly (e.g., write locks, retries) to preserve consistency.

Therefore, a **CP configuration** is ideal for such systems, ensuring that files remain consistent and reliable even if some nodes become temporarily unreachable.

Question 12: Online Voting System

An online voting platform is designed to accept millions of votes in real-time during a national election. Data accuracy is critical, and each vote must be recorded exactly once. Using CAP Theorem, justify the preferred system configuration among AP, CP, and CA.

An online voting system must prioritize **Consistency (C)** — no duplicate or lost votes can be tolerated. Given the potential for server failures or network issues during a nationwide election, **Partition Tolerance (P)** is necessary.

Availability can be compromised temporarily to ensure accurate vote recording.

Hence, a **CP configuration** is preferred. It guarantees accurate and consistent results across partitions, even if that means slowing down the service or rejecting some requests during high contention.