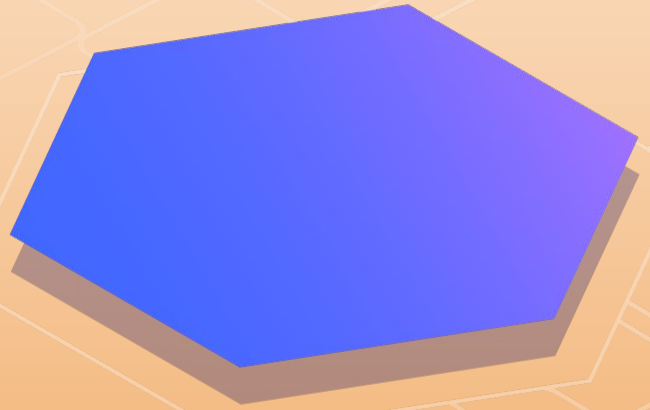
CAP Twelve Years Later: How the “Rules” Have Changed



Ikechi Iwuagwu

CAP THEOREM

- Any networked shared-data system can have only two of three desirable properties.

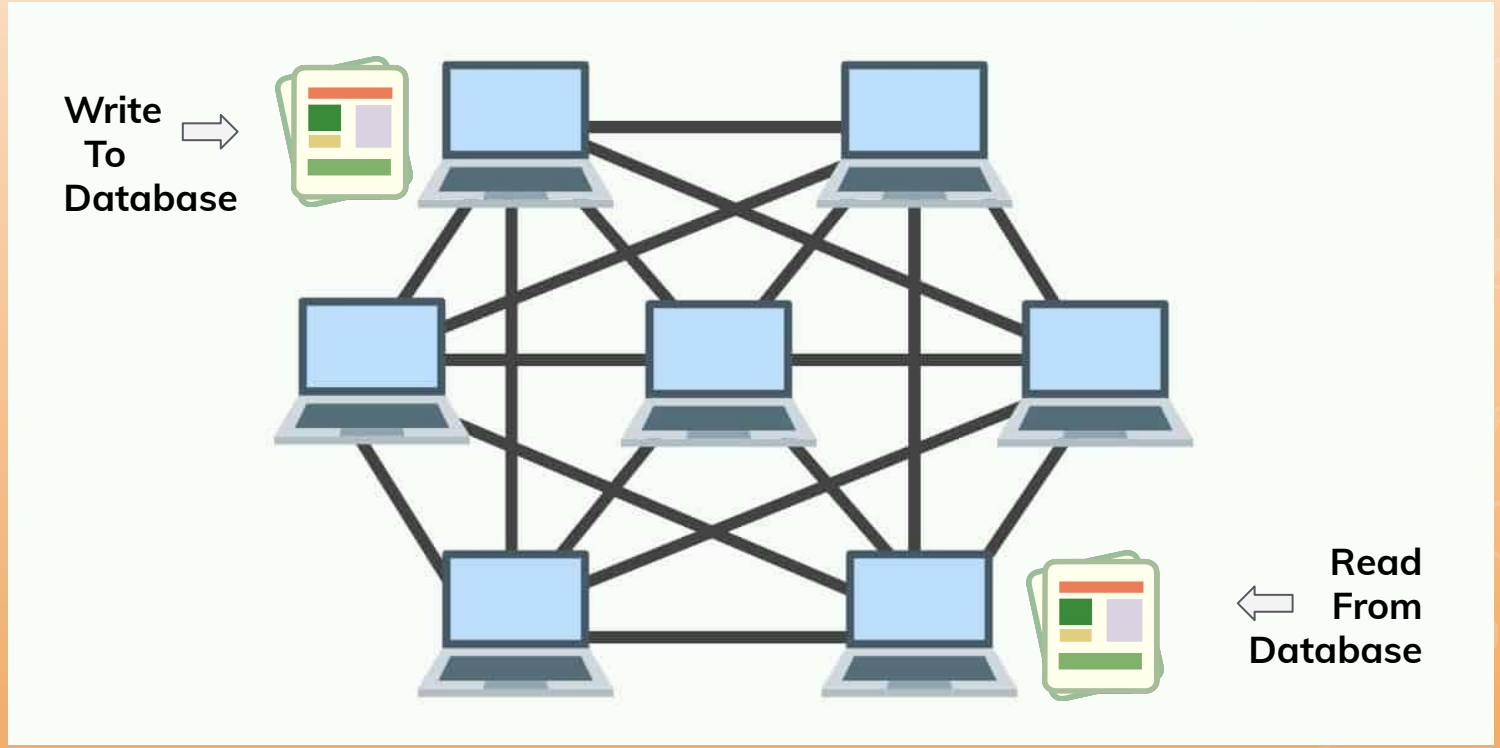
- CAP - The 3 Properties of a Distributed System
  - CONSISTENCY
  - AVAILABILITY
  - PARTITION TOLERANCE

CONSISTENCY

- If I write some data to the database from one node, and attempt to read it from another node, I should get back exactly what was just written or anything newer.

- The most up to date data.

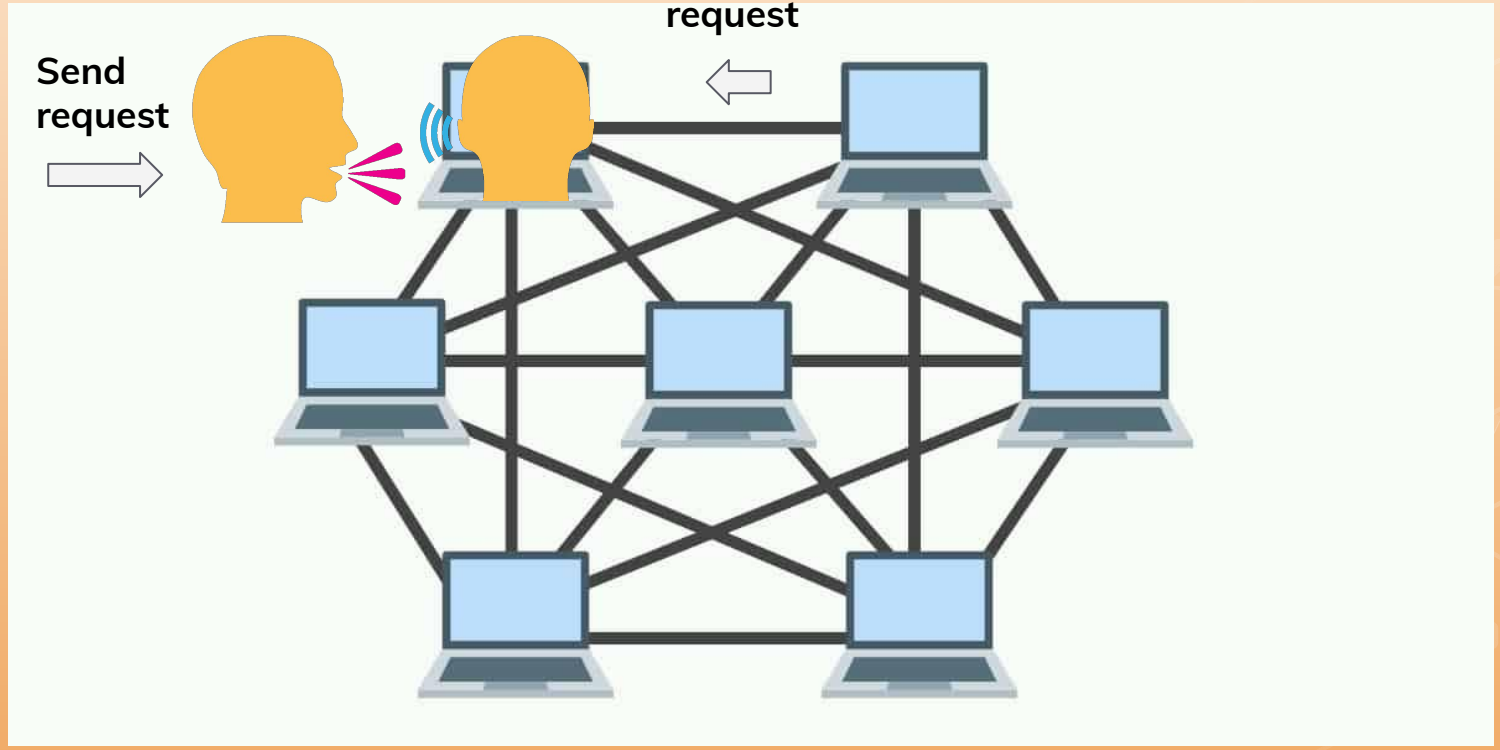
CONSISTENCY



AVAILABILITY

- If I attempt to communicate with one node, it should respond assuming that it has not failed.

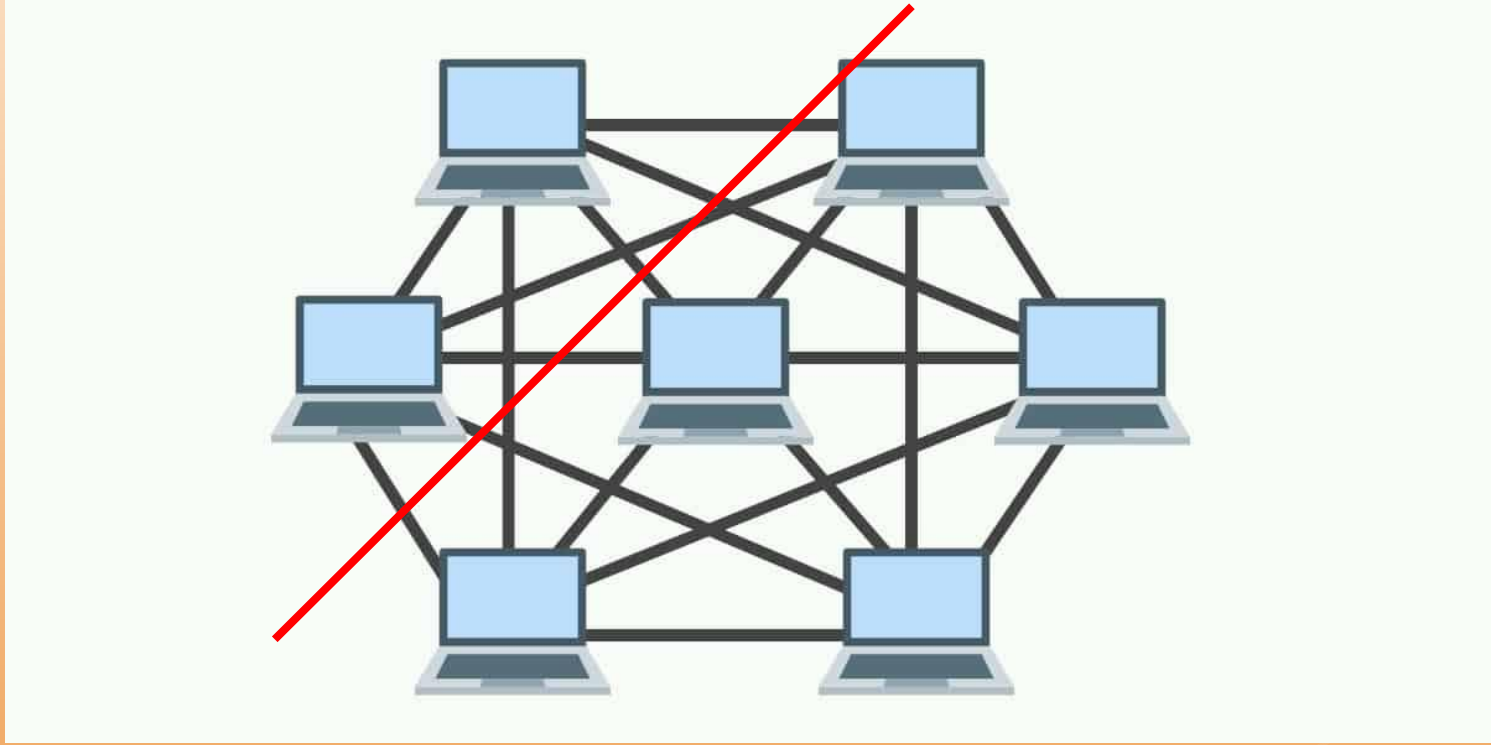
AVAILABILITY



PARTITION TOLERANCE

- The network should be able to be partitioned while still maintaining consistency and availability.

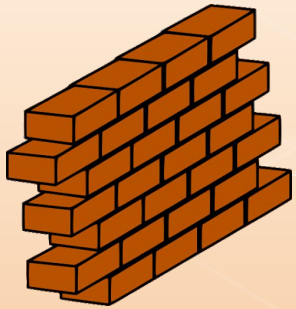
PARTITION TOLERANCE



CAP THEOREM

- The “2 of 3” formulation of the CAP Theorem says that we can only have at most 2 of 3 of these attributes.

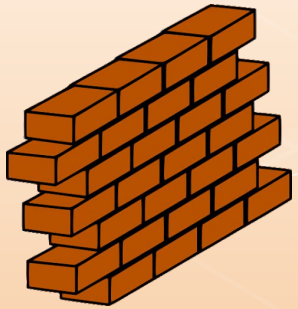
CAP THEOREM



Option 1: Consistent & Partition Tolerant

- If we want consistency and partition tolerance, we must sacrifice availability.

CAP THEOREM



Option 2: Available & Partition Tolerant

- If we want availability and partition tolerance, we must sacrifice consistency.

CAP THEOREM

Option 3: Consistent & Available

- If we want consistency and availability, we must sacrifice partition tolerance.

CAP THEOREM

- The general belief is that for wide-area systems, designers cannot forfeit partition tolerance.

- There are a number of reasons why one part of a network may not be able to communicate with the other.

CAP THEOREM

- In some ways, the NoSQL movement is about creating choices that focus on availability first and consistency second.

- Databases that adhere to ACID properties focus on consistency first.

Atomicity (A)

Consistency (C)

Isolation (I)

Durability (D)

CAP THEOREM

However...this is misleading.

CAP THEOREM

- CAP only prohibits perfect availability and consistency in the presence of partitions.

- The goal should be to maximize the combination of availability, consistency, and partition tolerance.

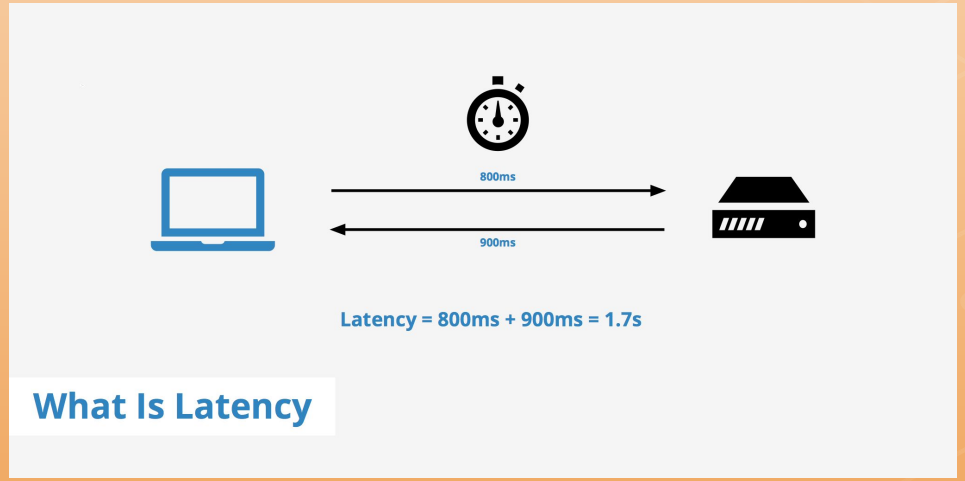
CAP THEOREM

- Partitions are a rare occurrence.

- The choice between consistency and availability does not have to be the same across different subsystems, operations, or types of data.

CAP-LATENCY CONNECTION

- The CAP theorem ignores latency, although latency and partitions are closely related in practice.



STRATEGY FOR PARTITIONS

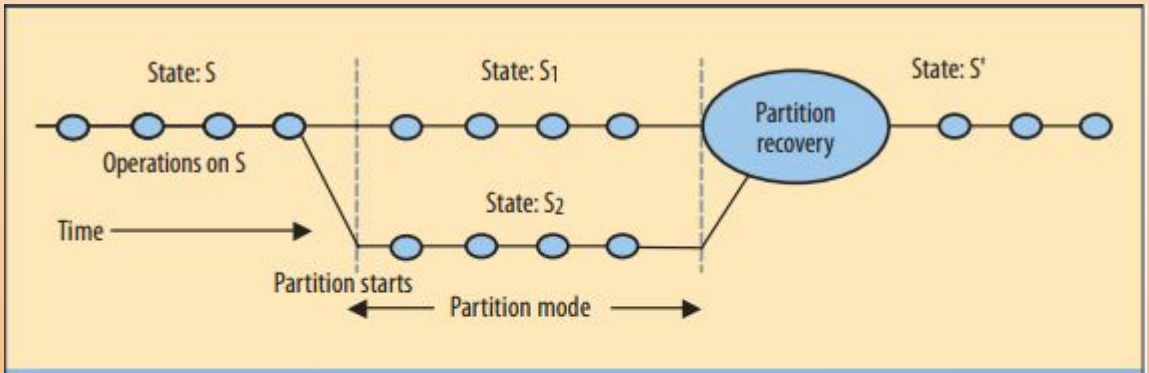


Figure 1. The state starts out consistent and remains so until a partition starts. To stay available, both sides enter partition mode and continue to execute operations, creating concurrent states S_1 and S_2, which are inconsistent. When the partition ends, the truth becomes clear and partition recovery starts. During recovery, the system merges S_1 and S_2 into a consistent state S′ and also compensates for any mistakes made during the partition.

PARTITION MODE STRATEGIES

Option 1: Limit some operations, thereby reducing availability.

Option 2: Record extra information about the operations that will be helpful during partition recovery.

WHICH OPERATIONS SHOULD PROCEED?

Allow duplicate primary keys during a partition, and fix later?

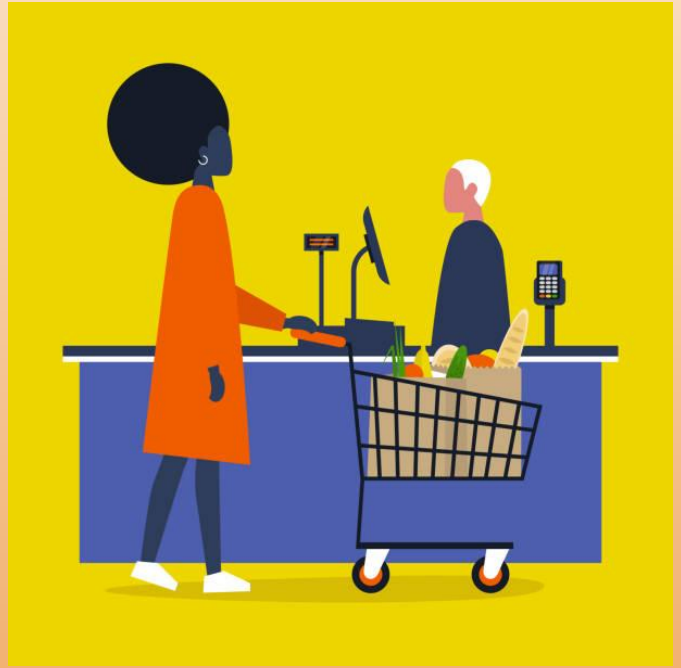
Primary Keys

StudentId	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

WHEN POLICIES MUST BE MAINTAINED

Having violated policies during a credit card transaction is generally a bad idea.

Instead, stop the operation and keep it in the order-processing state, until partition is resolved.



Version Vectors

- Elements are a pair (node, logical time) with one entry for every node that has updated the object and the time of its last update.

- Given two versions of an object, A and B, A is newer than B if, for every node in common in their vectors, A's times are greater than or equal to B's and at least one of A's times is greater.

PARTITION RECOVERY

The designer must solve two problems during recovery from a partition.

- The state on both sides must become consistent.

- There must be compensation for the mistakes made during partition mode.

COMMUTATIVE OPERATIONS

- Using commutative operations is the closest approach to a general framework for automatic state convergence.

- However, using only commutative operations is difficult.

COMMUTATIVE OPERATIONS

- Marc Shapiro and colleagues at INRIA have greatly improved the use of commutative operations for state convergence.

- They have developed commutative replicated data types (CRDTS), a class of data structures that provably converge after a partition.

FIXING MISTAKES

- Last Writer Wins
- Merge operations
- Human escalation

QUESTIONS?

Sources

CAP Twelve years later: How the "Rules" have Changed

Images

- https://www.comparitech.com/net-admin/network-topologies-advantages-disadvantages/
- https://www.clipartmax.com/middle/m2i8d3b1H7Z5N4d3_similar-clip-art-document-clipart/
- https://www.kindpng.com/picc/b/251-2511051_listen-png.png
- https://webcomicms.net/sites/default/files/clipart/131824/consistent-cliparts-131824-5038603.jpg
- http://www.i2clipart.com/clipart-red-brick-wall-fd61
- https://www.pinclipart.com/downpngs/bxxRwb_ok-icon-availability-available-icon-png-clipart/
- https://www.keycdn.com/img/support/what-is-latency-1.png
- http://rdbms.opengrass.net/2_Database%20Design/2.1_TermsOfReference/r/keyPrimary.gif
- https://www.istockphoto.com/vector/cashier-grocery-store-a-client-buying-groceries-at-the-supermarket-register-counter-gm1097907978-294850742