IMPORTANCE OF AVAILABILITY AND PARTITIONING AS COMPARED TO CONSISTENCY IN FACEBOOK,GOOGLE AND OTHER SOCIAL PLATFORMS

TEAM MEMBERS:

Likith Garapati -19BCD7245
Chaturya. Chinta - 19BCE7528
Aakash Challagundla -19BCI7077
Bhuvana.kommineni -19BCE7764
Anil kavuri - 19BCI7051

Introduction:

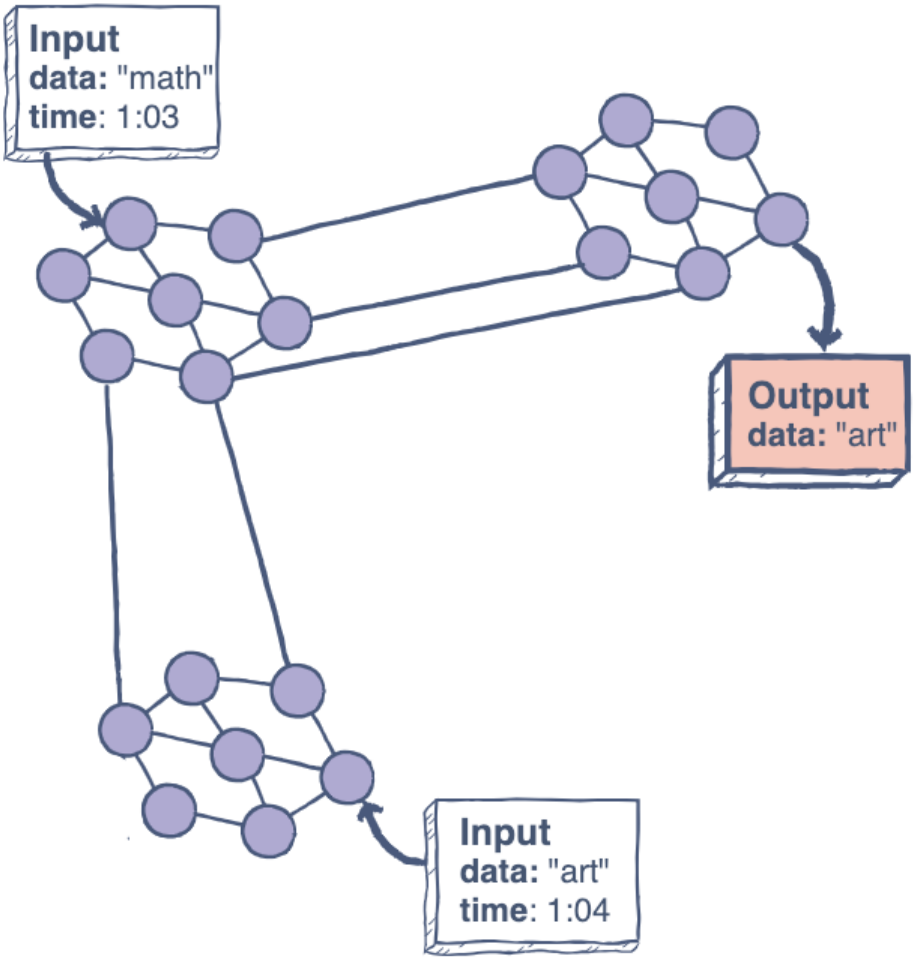
CAP theorem : The CAP theorem it is also called Brewer's theorem states that a distributed database system can only guarantee two out of these three characteristics:

1.Consistency
2. Availability
3. Partition Tolerance.

1.Consistency :

A system is said to be consistent if all nodes see the same data at the same time.

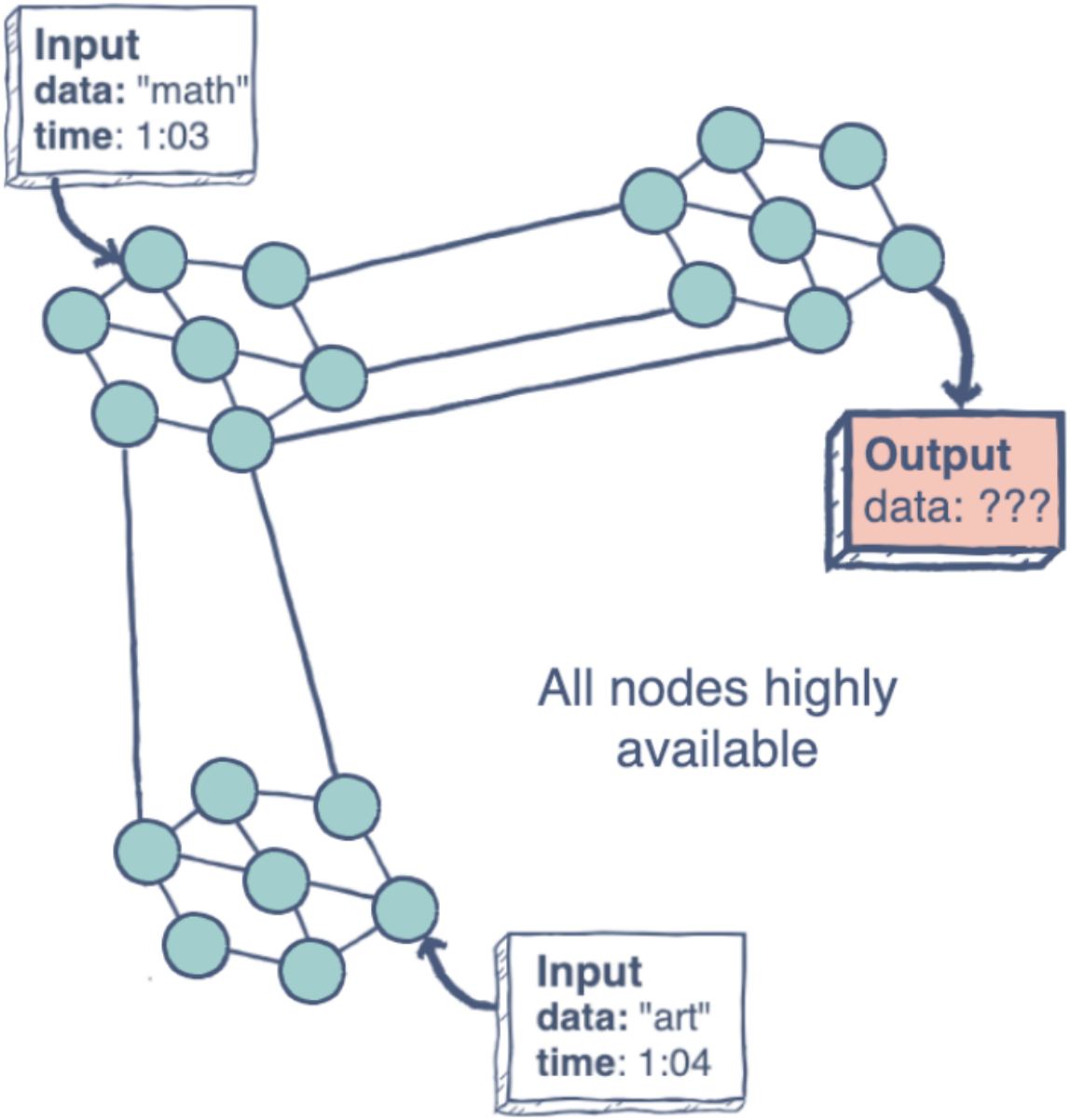
Simply, if we perform a read operation on a consistent system, it should return the value of the most recent write operation. This means that the read should cause all nodes to return the same data, i.e., the value of the most recent write.



2.Availability:

Availability in a distributed system ensures that the system remains operational 100% of the time. Every request gets a (non-error) response regardless of the individual state of a node.

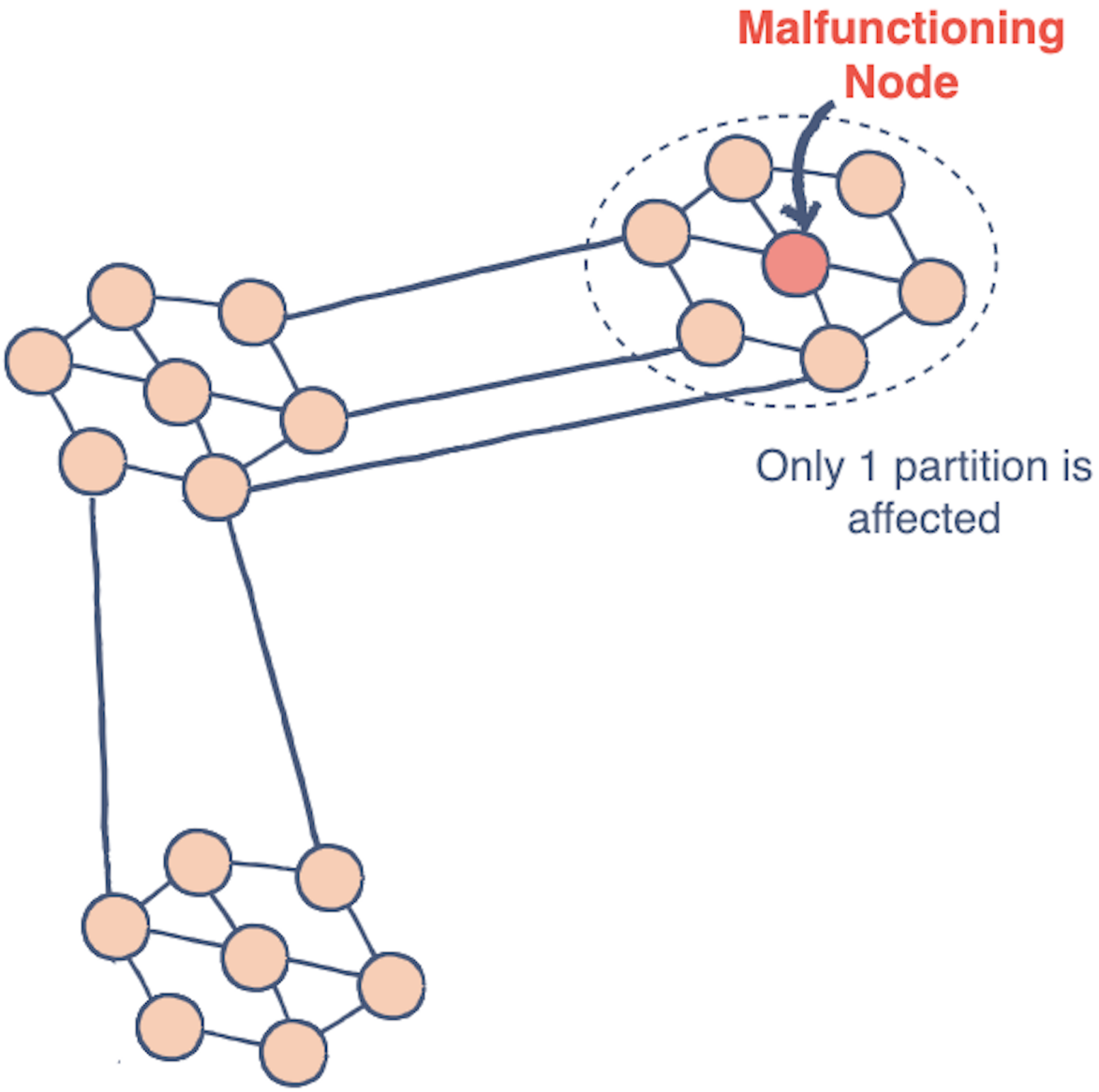
Ability to retrieve data that we store in distributed systems no matter what happens inside the cluster.



3.Partitionality Tolerance:

This condition states that the system does not fail, regardless of if messages are dropped or delayed between nodes in a system.

Partition tolerance has become more of a necessity than an option in distributed systems. It is made possible by sufficiently replicating records across combinations of nodes and networks.



Executive summary:

In distributed systems only two properties can be guaranteed at the same time.Either one of availability/partition tolerance/consistency is always compromised.

Case1: Availability is compromised when consistency and partition tolerance are preferred.
In case of expected network failure.Some systems prefer consistency in case of delay in update.The system will show an error message instead of answering queries.

Case2: Consistency is compromised when Availability and partition tolerance are preferred.
Instead of sending error messages this system will prefer sending un updated/previous data.

Case3: partition tolerance is compromised when Availability and Consistency are preferred.
Usually we cannot avoid partition tolerance in most of the cases because network problems can be expected or unexpected.This case can only be observed in non distributed systems.

Necessities of Consistency , Availability and partionality Tolerance in Real world Applications

1.All the three Consistency ,Availability and Partionality are equally important in the real world Scenario.

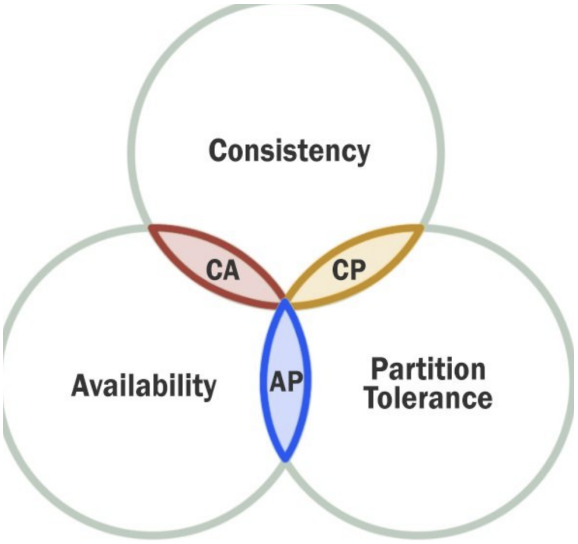
Consistency: All clients see the latest data even in the case of updates.

Availability: All clients can find a replica of some data even in the case of a node failure. This means that even if some part of the system goes down, the clients can still access the data.

Partition tolerance:

1. The system continues to work regardless of arbitrary message loss or failure of part of the system.
2. Depending upon the platforms and its requirements they give preference to any of the two properties

3. In the similar way Facebook,Instagram,linkedin,Google...so on all the social platforms give more priority to Availability and Partionality Tolerance by sacrificing Consistency.



Analysis:

Case Study on Availability and partionality Tolerance:

Facebook :

1) Facebook has Billions of users using every minute
2) If they compromised Availability for the sake of any of the Consistency they may lose their users
3) That's the reason why they prefer Availability.

Why Availability is given more priority than consistency the below data will explain

The moment in question is the user query. We assume that a user makes a query to a database, and the networked database is to return a value.

Whichever value is returned from the database depends on our choice to provide consistency or availability. Here's how this choice could play out:

On a query, we can respond to the user with the current value on the server, offering a highly available service. If we do this, there is no guarantee that the

value is the most recent value submitted to the database. It is possible a recent write could be stuck in transit somewhere.

If we want to guarantee high consistency, then we have to wait for the new write or return an error to the query. Thus, we sacrifice availability to ensure the data returned by the query is consistent.

Google*:*
Google Search focuses on availability and relaxes consistency - the results you see will depend on the state of the server that responds to your request, and different servers can have inconsistent states.
Communication tools like GMail and Facebook typically enforce only eventual consistency. Messages may take a while to travel between different servers, and so different servers may temporarily have inconsistent views of the world depending on what messages they have seen so far. However, all servers will eventually have consistent states, since the correct state is defined by interpreting the messages in the order of their global timestamps, rather than the order in which they arrived.

Instagram:

Just like all the other social platforms Instagram also focuses on Consistency and partition tolerance rather than consistency.

There are different profiles like professional,personal and business profiles too with a lot of data including new profiles,posts,stories etc being generated.

Databases:

Instagram mainly uses two backend database systems:PostgreSQL and Cassandra

In the case of instagram it leans towards eventual consistency or weak consistency. Instagram focuses on availability and partition tolerance.

case:Posts immediately don’t need to be updated on every feed.They can take some time or the website/app needs to be refreshed to get the updated data.This is known as eventual consistency or weak consistency.

Partition tolerance: Among different servers even if a problem exists in one data center/node it shouldn't be affecting the performance of the application in different countries/places.The app should still be up and running no matter what.Most of the social platforms give high priority to partition tolerance.

Availability: Instagrams showcase old/un updated data in spite of giving preference to consistency.

Conclusion:

1) We all have to choose between a system of properties: We have to choose only any two among the three systems.

2) No perfect system exists: That means no system can acquire all the three properties at the same, We have to sacrifice at least one property

3)Requirements of the System: We should always build a system based on the requirements which helps in the development of the system

4) Balance among relations Consistency,Availability and partionality: Balance among consistency,Availability,Consistency to be maintained by the system

*Spanner is google's highly available global scale distributed database

*It provides strong consistency for all transactions

*The combination of availability and consistency while over a wide an area is generally considered as impossible due to CAP theorem

* spanner achieves this combination and why it is consistent with CAP.