Availability

You want to book an urgent train ticket to your hometown. You login into the IRCTC website to book the ticket. The server is already overloaded because of chhath puja next week. You are trying to access the site for the last two hours, but the site is down or unavailable.

How will you feel? Is this a good User Experience? Is this reliable?

This is where availability comes into play. It is an essential system aspect and performance measure of the application. Let's try to understand it better.

What is availability?

Availability is the probability of whether a system will work as needed when the user wants to make a request. If an application responds every time the user makes a request, then the application is an available application.

Example: Google has a very supportive system and is 100% available/very high availability.

Availability and Architecture

Monolithic Architecture:

- A single centralised system with all layers implemented in one codebase, Monolithic Architecture can be deployed on different machines but data reliability decreases.
- In case of a fault in one unit of the system, the complete machine goes down.
- Monolithic architecture is prone to a Single Point Of Failure(SPOF).
- During such cases, the machine would be unavailable.

Monolithic Architecture => Low availability (due to SPOF)

Distributed System:

- Multiple machines are connected over a network.
- Has a mechanism to avoid Single Point Of Failure(SPOF).
- Due to redundancy and replication, the availability increases.

Distributed System => High availability (due to Redundancy/Replication)

Therefore,

Availability of Distributed Systems > Availability of Monolithic Architecture.

Fault Tolerance/Partition Tolerance

Failures are frequent in distributed systems. During a failure, the system should not go down, and the system must handle the fault gracefully.

Distributed system handle failures using:

1. Redundancy
2. Replication

If the machine is fault-tolerant, the availability would be higher. Because in case of failure, there is a duplicate machine to operate in place.

Therefore, Fault tolerance is directly proportional to availability.

Fault Tolerance $\propto$ Availability

How to increase the availability?

We can increase the availability of an application using the following ways:

1. Eliminate single points of failure:
   Use of replication or redundancy for avoiding Single Points of Failure.
   Example: We can replicate very hardware devices like two routers, two switches, two servers, two power sources instead of one.
2. Ensure Automatic Failover:
   Automatic Failover is automatically moving a machine as standby during a failure to preserve its uptime. It is heavily used by Amazon Web Services, Google Cloud, Microsoft Azure etc.
3. Implement Geographic Redundancy:
   Hosting the server at different geographical locations preferably close to the user,always to have, increases network performance. Hosting on multiple servers provides high availability along with better performance.
4. Keep Improving and Updating:
   Scripted deployments can be used for automatically updating the server. This allows users to always have access to the latest and most efficient version with more reliable solutions to security vulnerabilities thereby increasing the availability.
5. Provide excellent support:

The Data Storage and Cloud Service Providers should be highly accessible and up 24*7. Extremely responsive support can help increase availability.