

[Consistency Model in Distributed System]

Strong Consistency

- **Definition:** Ensures that all clients see the same data simultaneously after any update. The system behaves as if there is a single copy of the data.
- **Guarantee:** Once a write is completed, subsequent reads return the updated value.
- **Use Cases:** Financial transactions, banking systems, or any application requiring strict correctness.
- **Example Systems:** RDBMS, Spanner (Google).

Eventual Consistency

- **Definition:** Guarantees that, given enough time and no new updates, all replicas converge to the same state.
- **Guarantee:** Writes eventually propagate to all replicas, but reads may temporarily return stale data.
- **Use Cases:** Systems prioritizing availability and partition tolerance, such as social media feeds or DNS.
- **Example Systems:** DynamoDB, Cassandra.

Characteristics of Weak Consistency

1. **Low Synchronization:** Writes are not guaranteed to be immediately visible to all clients.
2. **High Performance:** By reducing synchronization overhead, weak consistency supports higher throughput and lower latency.
3. **Eventual Convergence (Usually):** Often, systems with weak consistency ensure that replicas eventually synchronize, though there's no strict timeline for this.