# 一致性

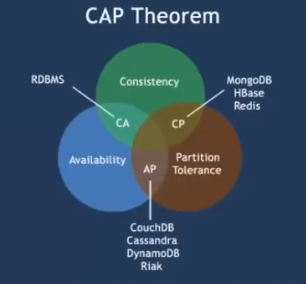
## CAP理论

对于一个分布式系统，不能同时满足以下三点：

一致性（Consistency）

可用性（Availability）

分区容错性（Partition Tolerance）



## 分类

### 弱一致性

#### 最终一致性

最终一致性：往分布式数据库中写入数据，此时另一个节点读数据，系统是无法保证能够读到最新的数据的，但是可以保证将来某一时刻可以读到写入的数据。

注：中兴GoldenDB基于最终一致性，阿里OceanBase基于强一致性。

##### DNS

DNS（Domain Name System）

##### Gossip

Gossip（Cassandra的通信协议）

### 强一致性

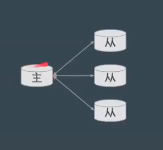
强一致性主要是解决这一问题：数据不能存在单点上。

分布式系统对于fault tolerance的一般解决方案是state machine replication（可以理解为一个函数，具有一个初始状态和最终状态，对应的状态就是日志log），严格地讲是state machine replication的共识（consensus）算法。Paxos其实是一个共识算法，系统的最终一致性不仅仅需要达成共识，还会取决于client的行为。

#### 主从同步

主从同步复制：

1. Master接受写请求；
2. Master复制日志到slave
3. Master等待，直到**所有**从库返回



存在问题：

一个节点失败，Master阻塞，导致整个集群不可用，保证了一致性，可用性却大大降低。

基本想法：多数派

每次写都保证写入大于N/2个节点，每次读保证从大于N/2个节点中读

但是，在并发环境下，多数派无法保证系统正确性，顺序非常重要！需要引入共识算法。



#### Paxos

#### Raft

#### ZAB