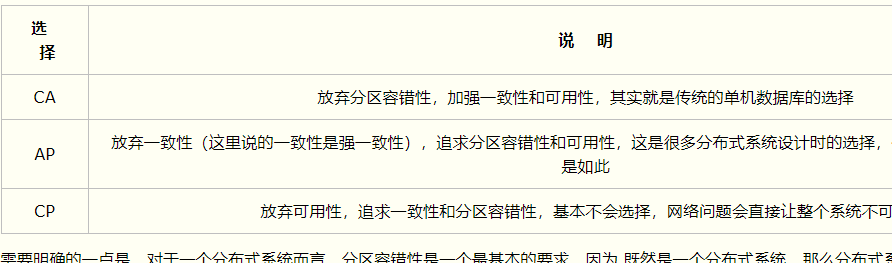
* [从ACID到CAP到BASE](https://segmentfault.com/a/1190000004468442" \t "_blank)
* [2PC到3PC到Paxos到Raft到ISR](https://segmentfault.com/a/1190000004474543)
* [复制、分片和路由](https://segmentfault.com/a/1190000004485355)
* [副本更新策略](https://segmentfault.com/a/1190000004480546)
* [负载均衡算法及手段](https://segmentfault.com/a/1190000004492447)
* [RWN及Quorum与强一致性](https://segmentfault.com/a/1190000004499551)

Cap理论：

一致性，可用性，容错性，其经典理论就是最好的情况的就是只能满足其中两个条件，我不信后来使用归纳法，发现确实，

因为一致性中，存在绝对的一致性和最终一致性，绝对一致性如果想要实现，就必须只有一个数据库，只有一个数据库自然就满足不了容错，因为一旦宕机，服务就不可用。最后妥协到最终一致性。因为强一致性使得其它ap都不可用，那么就妥协一点，最终一致性，而最终一致性就意味着需要副本。最终一致性是一种妥协，其实还是会存在不一致的问题。当然有人会说，可以等他们都同步完了在提供服务，这个时候不就违背了服务可用性么！要看场景更关注哪一个方面：



对于服务可用仿照最终一致性的妥协，提出了基本可用的概念，允许部分功能暂时处于不可用或者延迟，引导到一个降级页面。

这种妥协我们称其为base理论，base理论中的s称其为软状态，我觉得在某种程度上跟最终一致性有些冲突。

* [2PC到3PC到Paxos到Raft到ISR](https://segmentfault.com/a/1190000004474543)

作为一致性和容错的实现方式，结果发展过程，基本以少数服从多数的方式成为主流。Paxos，zab，raft等等，这里说一个容错成本概念：

对于paxos和zab，如果要允许两个副本故障就必须有2f+1=5的副本，这就引出了kafka使用的isr方式，主集合才能成为最终的备选副本。有点像国家领导人的选举，只能从政府机构中选举，而不是平民

理论这些东西，考试的时候看是最好的，品尝用的比较少，只要了解即可。