

经历过技术面试的小伙伴想必对这个两个概念已经再熟悉不过了！

Guide哥当年参加面试的时候，不夸张地说，只要问到分布式相关的内容，面试官几乎是必定会问这两个分布式相关的理论。

并且，这两个理论也可以说是小伙伴们学习分布式相关内容的基础了！

因此，小伙伴们非常非常有必要将这理论搞懂，并且能够用自己的理解给别人讲出来。

这篇文章我会站在自己的角度对这两个概念进行解读！

个人能力有限。如果文章有任何需要改善和完善的地方，欢迎在评论区指出，共同进步！——爱你们的Guide哥

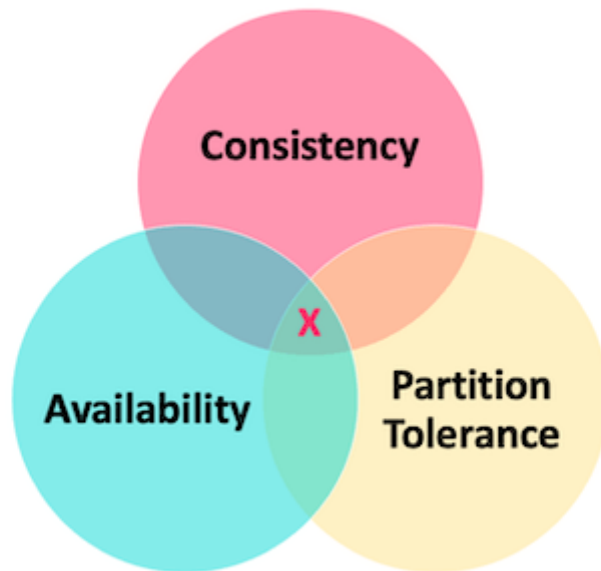
## CAP理论

**CAP 理论/定理**起源于 2000 年，由加州大学伯克利分校的Eric Brewer教授在分布式计算原理研讨会（PODC）上提出，因此 CAP定理又被称作 **布鲁尔定理（Brewer's theorem）**

2年后，麻省理工学院的Seth Gilbert和Nancy Lynch 发表了布鲁尔猜想的证明，CAP理论正式成为分布式领域的定理。

### 简介

**CAP** 也就是 **Consistency（一致性）、Availability（可用性）、Partition Tolerance（分区容错性）** 这三个单词首字母组合。



CAP 理论的提出者布鲁尔在提出 CAP 猜想的时候，并没有详细定义 **Consistency、Availability、Partition Tolerance** 三个单词的明确定义。

因此，对于 CAP 的民间解读有很多，一般比较被大家推荐的是下面  这种版本的解。

在理论计算机科学中，CAP 定理（CAP theorem）指出对于一个分布式系统来说，当设计读写操作时，只能能同时满足以下三点中的两个：

- **一致性（Consistence）**：所有节点访问同一份最新的数据副本
- **可用性（Availability）**：非故障的节点在合理的时间内返回合理的响应（不是错误或者超时的响应）。
- **分区容错性（Partition tolerance）**：分布式系统出现网络分区的时候，仍然能够对外提供服务。

### 什么是网络分区？

分布式系统中，多个节点之前的网络本来是连通的，但是因为某些故障（比如部分节点网络出了问题）某些节点之间不连通了，整个网络就分成了几块区域，这就叫网络分区。



### 不是所谓的“3 选 2”

大部分人解释这一定律时，常常简单的表述为：“一致性、可用性、分区容忍性三者你只能同时达到其中两个，不可能同时达到”。实际上这是一个非常具有误导性质的说法，而且在 CAP 理论诞生 12 年之后，CAP 之父也在 2012 年重写了之前的论文。

当发生网络分区的时候，如果我们要继续服务，那么强一致性和可用性只能 2 选 1。也就是说当网络分区之后 P 是前提，决定了 P 之后才有 C 和 A 的选择。也就是说分区容错性（Partition tolerance）我们是必须要实现的。

简而言之就是：CAP 理论中分区容错性 P 是一定要满足的，在此基础上，只能满足可用性 A 或者一致性 C。

因此，分布式系统理论上不可能选择 CA 架构，只能选择 CP 或者 AP 架构。

### 为啥无同时保证 CA 呢？

举个例子：若系统出现“分区”，系统中的某个节点在进行写操作。为了保证 C，必须要禁止其他节点的读写操作，这就和 A 发生冲突了。如果为了保证 A，其他节点的读写操作正常的话，那就和 C 发生冲突了。

选择的关键在于当前的业务场景，没有定论，比如对于需要确保强一致性的场景如银行一般会选择保证 CP。

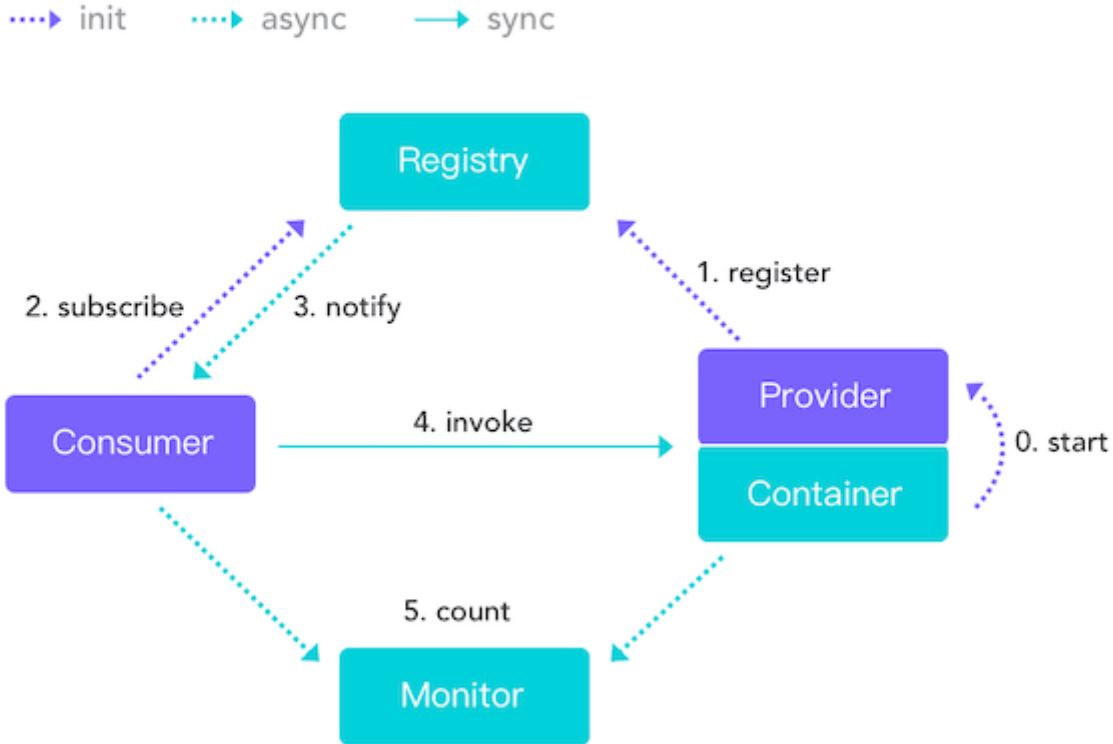
### CAP 实际应用案例

我这里以注册中心来探讨一下 CAP 的实际应用。考虑到很多小伙伴不知道注册中心是干嘛的，这里简单以 Dubbo 为例说一说。

下图是 Dubbo 的架构图。注册中心 Registry 在其中扮演了什么角色呢？提供了什么服务呢？

注册中心负责服务地址的注册与查找，相当于目录服务，服务提供者和消费者只在启动时与注册中心交互，注册中心不转发请求，压力较小。

## Dubbo Architecture



常见的可以作为注册中心的组件有：ZooKeeper、Eureka、Nacos....。

1. **ZooKeeper 保证的是 CP。** 任何时刻对 ZooKeeper 的读请求都能得到一致性的结果，但是，ZooKeeper 不保证每次请求的可用性比如在 Leader 选举过程中或者半数以上的机器不可用的时候服务就是不可用的。
2. **Eureka 保证的则是 AP。** Eureka 在设计的时候就是优先保证 A（可用性）。在 Eureka 中不存在什么 Leader 节点，每个节点都是一样的、平等的。因此 Eureka 不会像 ZooKeeper 那样出现选举过程中或者半数以上的机器不可用的时候服务就是不可用的情况。Eureka 保证即使大部分节点挂掉也不会影响正常提供服务，只要有一个节点是可用的就行了。只不过这个节点上的数据可能并不是最新的。
3. **Nacos 不仅支持 CP 也支持 AP。**

## 总结

在进行分布式系统设计和开发时，我们不应该仅仅局限在 CAP 问题上，还要关注系统的扩展性、可用性等等

在系统发生“分区”的情况下，CAP 理论只能满足 CP 或者 AP。要注意的是，这里的前提是系统发生了“分区”

如果系统没有发生“分区”的话，节点间的网络连接通信正常的话，也就不存在 P 了。这个时候，我们就可以同时保证 C 和 A 了。

总结：**如果系统发生“分区”，我们要考虑选择 CP 还是 AP。如果系统没有发生“分区”的话，我们要思考如何保证 CA。**

## 推荐阅读

1. [CAP 定理简化](#)（英文，有趣的案例）
2. [神一样的 CAP 理论被应用在何方](#)（中文，列举了很多实际的例子）
3. [请停止呼叫数据库 CP 或 AP](#)（英文，带给你不一样的思考）