

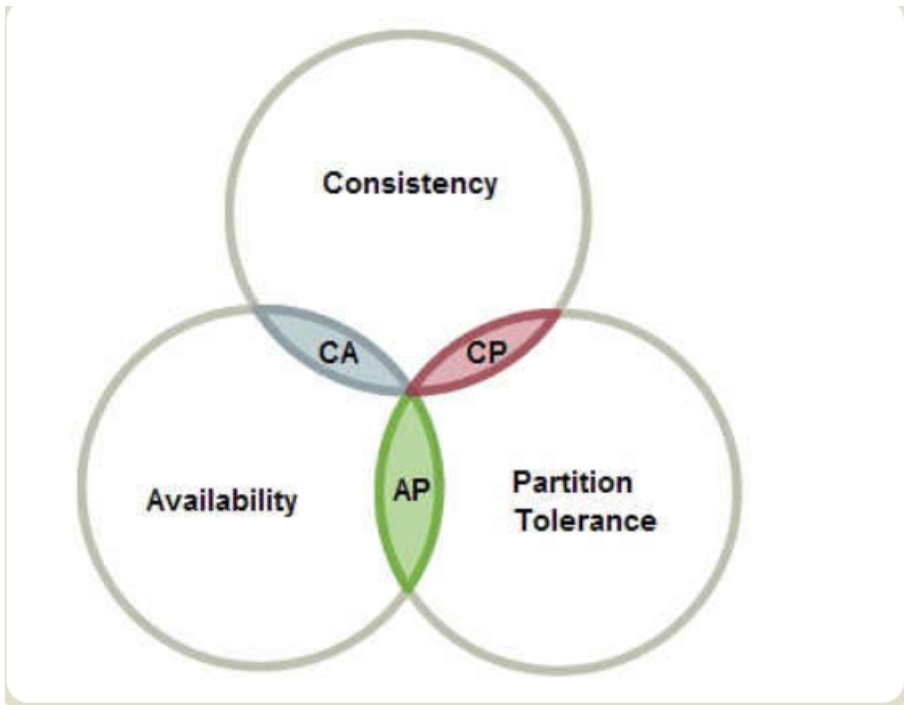
CAP理论

一:分布式系统的三个指标

1:Consistency (持久性)

2:Availability(可用性)

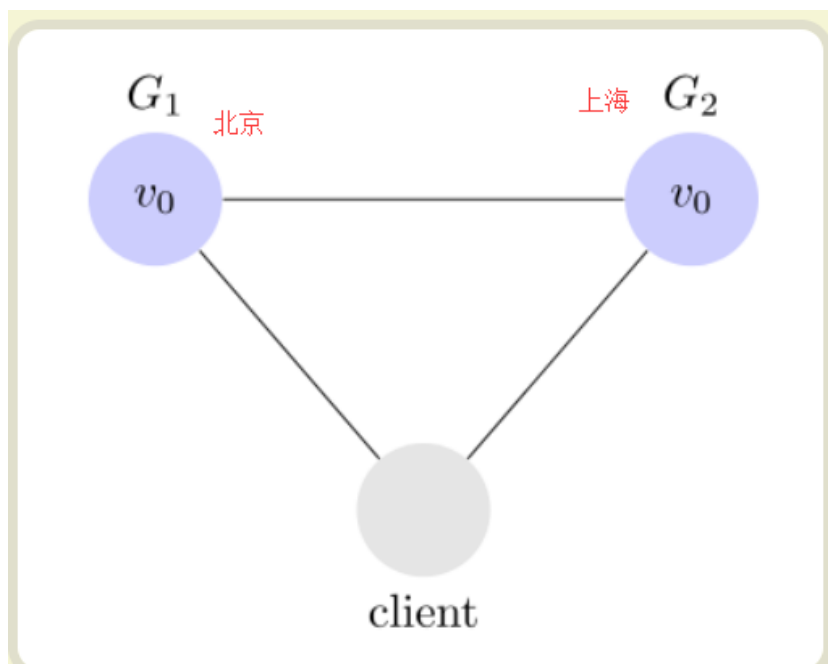
3:Partition tolerance(分区容错性)



结论:这三个指标不可能同时做到。这个结论就叫做 CAP 定理。

戏说分区容错性 (Partition tolerance)

大多数分布式系统都分布在多个子网络。每个子网络就叫做一个区 (partition)。分区容错的意思是，区间通信可能失败(**由于网路原因**)。比如，一台服务器放在北京，另一台服务器放在上海，这就是两个区，它们之间由于**网络抖动原因导致不能通信**。

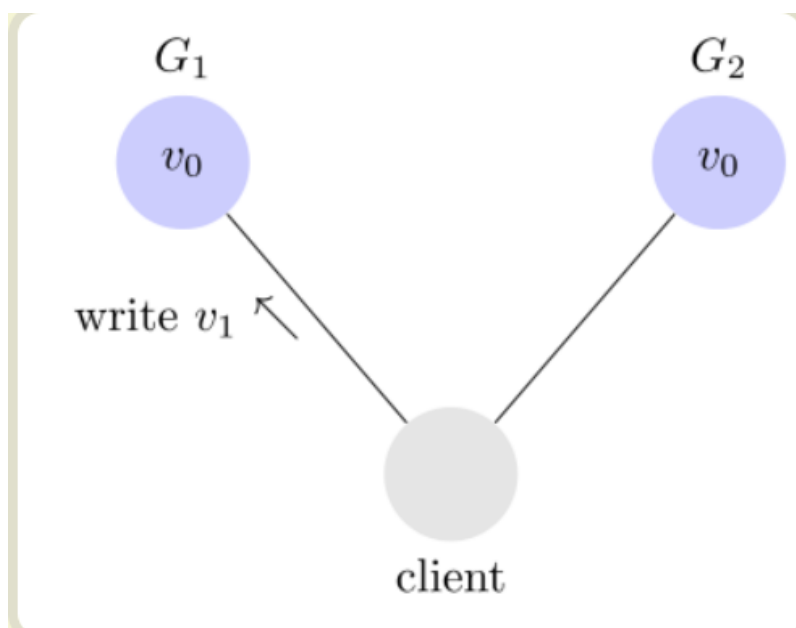


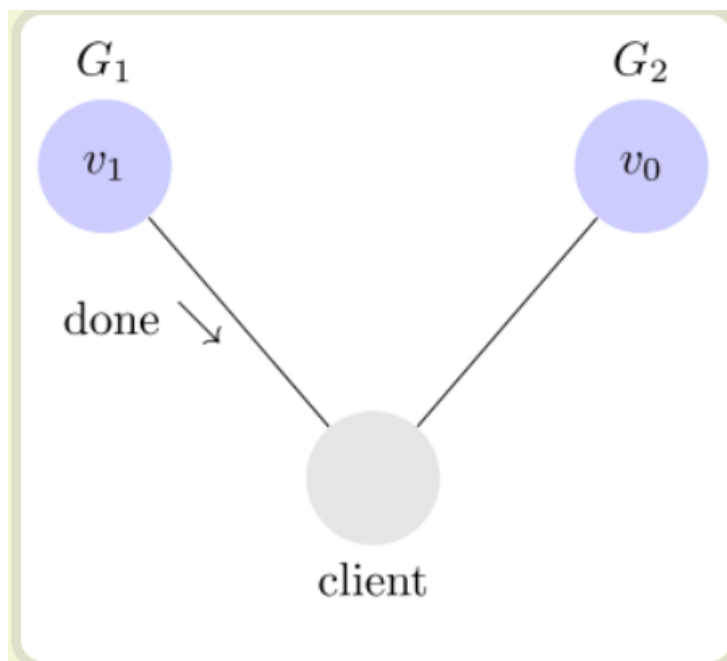
上图中， G_1 和 G_2 是两台跨区的服务器。 G_1 向 G_2 发送一条消息， G_2 可能无法收到。系统设计的时候，必须考虑到这种情况。

一般来说，分区容错无法避免，因此可以认为 CAP 的 P 总是成立。CAP 定理告诉我们，剩下的 C 和 A 无法同时做到。

Consistency (一致性) (各个时间上，各个服务器上的数据必须要一致的)

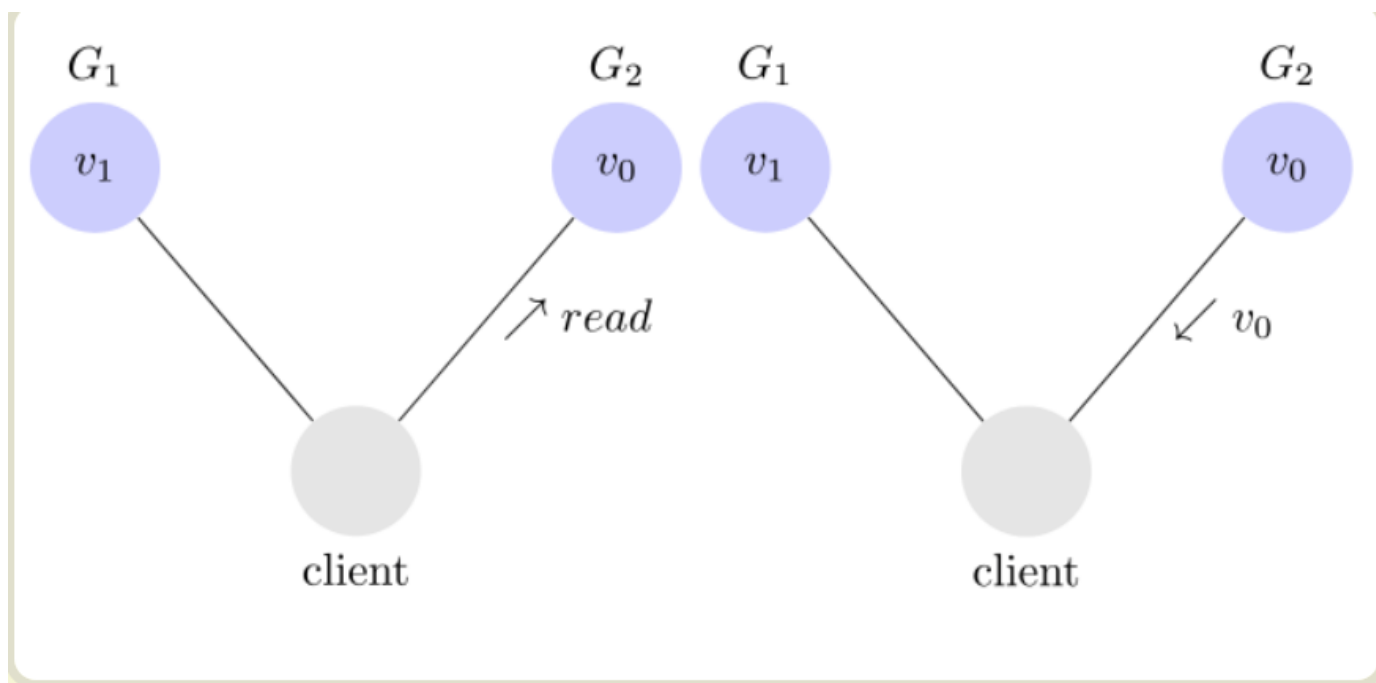
Consistency 中文叫做"一致性"。意思是，写操作之后的读操作，必须返回该值。举例来说，某条记录是 v_0 ，用户向 G_1 发起一个写操作，将其改为 v_1 。



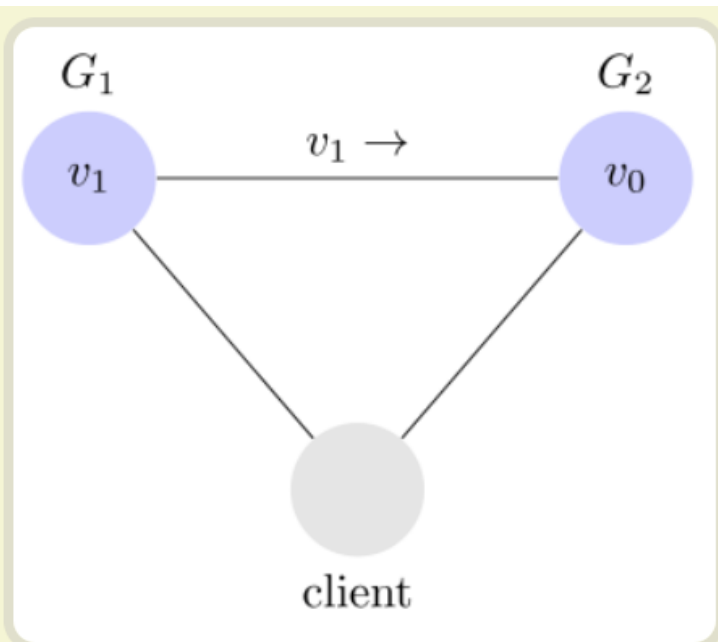


存在的问题:

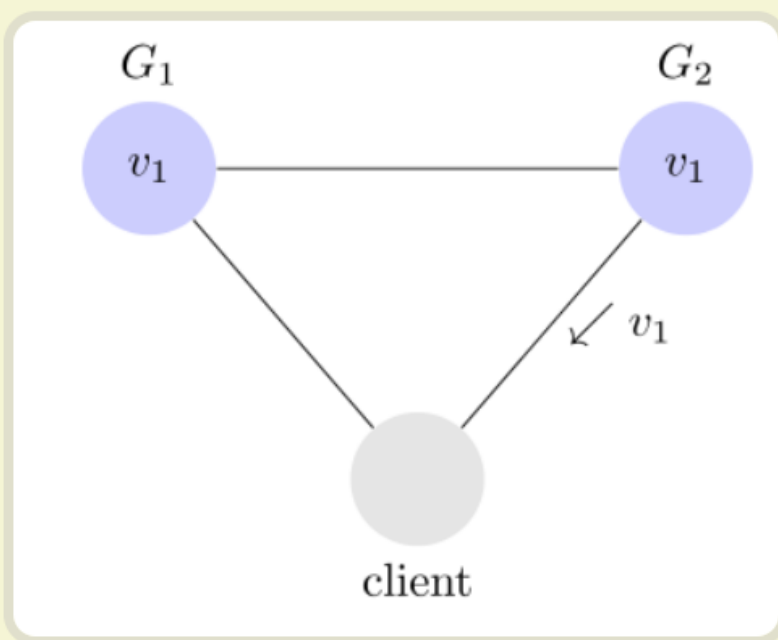
用户有可能向 G_2 发起读操作, 由于 G_2 的值没有发生变化, 因此返回的是 v_0 。 G_1 和 G_2 读操作的结果不一致, 这就不满足一致性了。



解决方案: 为了让 G_2 也能变为 v_1 , 就要在 G_1 写操作的时候, 让 G_1 向 G_2 发送一条消息, 要求 G_2 也改成 v_1 。



这样的话，用户向 G_2 发起读操作，也能得到 v_1 。



Availability 可用性

Availability 中文叫做“可用性”，意思是只要收到用户的请求，服务器就必须给出回应。

用户可以选择向 G_1 或 G_2 发起读操作。不管是哪台服务器，只要收到请求，就必须告诉用户，到底是 v_0 还是 v_1 ，否则就不满足可用性。

为啥CAP 只能三选二

Consistency 和 Availability 的矛盾

一致性和可用性，为什么不可能同时成立？答案很简单，因为可能通信失败（即出现分区容错）。

如果保证 G2 的一致性，那么 G1 必须在写操作时，锁定 G2 的读操作和写操作。只有数据同步后，才能重新开放读写。锁定期间，G2 不能读写，没有可用性不。

如果保证 G2 的可用性，那么势必不能锁定 G2，所以一致性不成立。

综上所述，G2 无法同时做到一致性和可用性。系统设计时只能选择一个目标。如果追求一致性，那么无法保证所有节点的可用性；如果追求所有节点的可用性，那就没法做到一致性。