



链滴

# 如何理解 CAP

作者: [skyesx](#)

原文链接: <https://ld246.com/article/1450014929848>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;"><span style="color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: 1; word-spacing: 0px; -webkit-text-stroke-width: 0px; display: inline !important; float: none; background-color: #ffffff;"> (本文是自己初期对CAP的肤浅认识，实际上CAP理论说不可能发生的情况只有“在分区发生的情况下，无法实现完美的一致性与可用性”&rdquo;，大家可以看我博文里转的一片文章，CAP理论1年回顾，以下内容有兴趣也可以喵喵，并无大致错误。) </span></p>

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;">&nbsp;&nbsp;&nbsp;</p>

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;">&nbsp;&nbsp;&nbsp;</p>

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;">CAP大家都知道，分别指代 Consistency, Availability, Partition tolerance。CAP theorem就是说 Consistency, availability, Partition tolerance最多同时只能满足两项，不可能三项都同时满足。<br style="background-color: inherit;" />虽然定理的描述很简单，但是其过于简单了，让人觉得模棱两可，下面我按自己对CAP的理解举例讲解一下把~</p>

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;">首先给定义把：</p>

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;">Consistency:</p>

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;">一致性 一致性有不同的层次，有不同的维度。但在本文中**最强的一致性**，简称**强一致性**指代：只要一个partition写入了某个更新，那么其他的partition就马上能读取得到。</p>

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;">Availability:</p>

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;">可用性

可用性也有不同的层次，不同的维度。本文中，最强的可用性指代，在应用功能相同的集群的所有机器里，每台机器都提供相同的功能，且只要有一台能用，那么就能往外提供完整的服务。

Partition tolerance:

分区容忍性。就是说即使集群内的机器无法互相通讯，也能对外提供服务。这个条件是由上述Consistency以Availability的实现方案中衍生出来的，直接看结论会比较难理解为啥会出现这项...

&nbsp;

下面我试用简单的文字来描述为何强一致性，强可用性及分区容忍性不能同时实现把：

1、假设我们有两台机器，两台机器都可读可写（完整的可用性），我们要如何实现强一致性？

我们很容易想到，应用程序写入到一台机器的时候，写入本地，通过网络发送变更到远程的机器，提交远程，交本地。那么我们的两台机器的集群就符合了强一致性。

2、好了一致性实现了，但如果一台机器A挂了，另外一台机器B也无法运转了，因为B无法将数据同步写入到，只能等到A恢复后才能继续工作。若抛下A不管，那么A在启动了之后，数据跟B就不一致了，没有证强一致性，那要怎么解决呢？聪明的你肯定已经有了想法了~

&nbsp;

B机器探测到A已经“确认A已经挂了”的时候，就把A没有同步的内容记到一个地方，等A重启的时候，读取B主机的更新，然后写本机，写入完成后，再跟B一起执行强一致性的协同操作。这时候，你就有了强一致性以及强可用了，嗯，感觉棒棒的

3、但回看一下上述解决方案中的加粗字段——“确认A已经挂了”

确认A已经挂了”，在实际的网络环境中，我们能确认A已经挂了吗？显然是不行的，有可能只是A与B之间的网络中断了！当A,B之间的网络中断后，如果A,B都认定这种情况是对方死掉了，然后各自记录自己的更新，等待对方来获取的时候就会发生冲突，产生不一致的情况（脑裂）！实际上机器A,B是无法判断远在哪方的TA是网络断了呢，还是TA挂了...因此结论就是...在脑裂的情况下 放弃可用性，等待两台机器互连开始，才提供服务，或者在脑裂的情况下 放弃一致性 保证可用性。

所以我为CAP最多只能实现两个而不能三者兼得 可以这么理解：

1、在不考虑脑裂的情况下，我们可以实现 强可用及强一致

2、在考虑脑裂的情况下，如果我们要求 强可用 那么就必须放弃 强一致

3、在考虑脑裂的情况下，如果我们要求 强一致 那么就必须放弃 强可用

在实际运用中，我们脑裂基本上是不允许发生的，如果发生了，那就是一场真正的灾难了。（在写这篇文章之前阅读了一篇新的文章，知识面得到了扩张，实际上分区的场景也是允许存在的，还有很多的系统实现见<http://www.infoq.com/cn/articles/cap-twelve-years-later-how-the-rules-have-changed>）那么就只能在 可用性 及 一致性 中作文章了...好在，可用性与一致性并非一个二选一的问题，要了可用性就不能要一致性，要了一致性就不能要可用性。

在考虑脑裂的情况下，一致性与可用性的关系可以用一个形容词来比喻 此消彼长。如果你要完全的可用性，那一一致性就变成了0，如果你要完全的一致性，那么可用性就变成了0。

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;">所以，  
们考虑设计分布式系统的时候，大多数情况下就是考虑 一致性 及 可用性 平衡。 </p>

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;">好在，  
人就给我们留下了关于这个平衡的思考，如BASE理念及PAXOS算法都可以看到CAP的影子，在之后博文里，我会继续介绍PAXOS算法以及BASE理念 </p>

<p style="margin: 5px 0px; background-color: #ffffff; color: #000000; font-family: 微软雅黑; font-size: 14px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: 21px; orphans: auto; text-align: left; text-indent: 0px; text-transform: none; white-space: normal; widows: auto; word-spacing: 0px; -webkit-text-stroke-width: 0px;">&nbsp;  </p>