## 1.1传统的ACID分别是什么？

### 1.1.1A（Atomicity）原子性

原子性很容易理解，也就是说事务里的所有操作要么全部做完，要么都不做，事务成功的条件是事务里的所有操作都成功，只要有一个操作失败，整个事务就失败，需要回滚。比如银行转账，从A账户转100元至B账户，分为两个步骤：1）从A账户取100元；2）存入100元至B账户。这两步要么一起完成，要么一起不完成，如果只完成第一步，第二步失败，钱会莫名其妙少了100元。

### 1.1.2C（Consistency）一致性

一致性也比较容易理解，也就是说数据库要一直处于一致的状态，事务的运行不会改变数据库原本的一致性约束。

### 1.1.3I（Isolation）独立性

所谓的独立性是指并发的事务之间不会互相影响，如果一个事务要访问的数据正在被另外一个事务修改，只要另外一个事务未提交，它所访问的数据就不受未提交事务的影响。比如现有有个交易是从A账户转100元至B账户，在这个交易还未完成的情况下，如果此时B查询自己的账户，是看不到新增加的100元的

### 1.1.4D（Durability）持久性

持久性是指一旦事物提交后，它所做的修改将会永久的保存在数据库上，即使出现宕机也不会丢失。

## 1.2CAP

### 1.2.1C：Consistency（强一致性）

对于任何从客户端发达到分布式系统的数据读取请求，要么读到最新的数据要么失败。换句话说，一致性是站在分布式系统的角度，对访问本系统的客户端的一种承诺：要么我给您返回一个错误，要么我给你返回绝对一致的最新数据，不难看出，其强调的是数据正确。

### 1.2.2A：Availability（可用性）

对于任何求从客户端发达到分布式系统的数据读取请求，都一定会收到数据，不会收到错误，但不保证客户端收到的数据一定是最新的数据。换句话说，可用性是站在分布式系统的角度，对访问本系统的客户的另一种承诺：我一定会给您返回数据，不会给你返回错误，但不保证数据最新，强调的是不出错。

### 1.2.3P：Partition tolerance（分区容错性）

分布式系统应该一直持续运行，即使在不同节点间同步数据的时候，出现了大量的数据丢失或者数据同步延迟。换句话说，分区容忍性是站在分布式系统的角度，对访问本系统的客户端的再一种承诺：我会一直运行，不管我的内部出现何种数据同步问题，强调的是不挂掉。

## 1.3经典CAP图

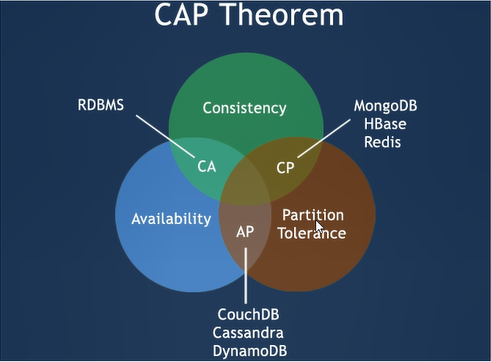
CAP理论的核心是：一个分布式系统不可能同时很好的满足一致性，可用性和分区容错性这三个需求，最多只能同时较好的满足两个。

因此，根据CAP原理将NoSQL数据库分成了满足CA原则、满足CP原则和满足AP原则三大类：

CA-单点集群，满足一致性，可用性的系统，通常在可扩展性上不太强大。

CP-满足一致性，分区容忍性的系统，通常性能不是特别高。

AP-满足可用性，分区容忍性的系统，通常可能对一致性要求低一些。



## 1.4CAP的3进2

（1）CAP理论就是说在分布式存储系统中，最多只能实现上面的两点。而由于当前的网络硬件肯定会出现延迟丢包等问题，所以分区容忍性是我们必须需要实现的。所以我们只能在一致性和可用性之间进行权衡，没有NoSQL系统能同时保证这三点（C:强一致性A：高可用性P：分布式容忍性）。

CA 传统Oracle数据库

AP 大多数网站架构的选择

CP Redis、Mongodb

（2）注意：分布式架构的时候必须做出取舍。一致性和可用性之间取一个平衡。多余大多数web应用，其实并不需要强一致性。因此牺牲C换取P，这是目前分布式数据库产品的方向。

（3）一致性与可用性的决择

对于web2.0网站来说，关系数据库的很多主要特性却往往无用武之地。

数据库事务一致性需求：

很多web实时系统并不要求严格的数据库事务，对读一致性的要求很低，有些场合对写一致性要求并不高。允许实现最终一致性。

数据库的写实时性和读实时性需求：

对关系数据库来说，插入一条数据之后立刻查询，是肯定可以读出来这条数据的，但是对于很多web应用来说，并不要求这么高的实时性，比方说发一条消息之后，过几秒乃至十几秒之后，我的订阅者才看到这条动态是完全可以接受的。

对复杂的SQL查询，特别是多表关联查询的需求：

任何大数据量的web系统，都非常忌讳多个大表的关联查询，以及复杂的数据分析类型的报表查询，特别是SNS类型的网站，从需求以及产品设计角度，就避免了这种情况的产生。往往更多的只是单表的主键查询，以及单表的简单条件分页查询，SQL的功能被极大的弱化了。

## 1.5BASE

BASE是什么？

BASE就是为了解决关系数据库强一致性引起的问题而引起的可用性降低而提出的解决方案。

BASE其实是下面三个术语的缩写：

基本可用（BasicallyAvailable）、软状态（Softstate）、最终一致（Eventuallyconsistent）

它的思想是通过让系统放松对某一时刻数据一致性的要求来换取系统整体伸缩性和性能上改观。为什么这么说呢，缘由就在于大型系统往往由于地域分布和极高性能的要求，不可能采用分布式事务来完成这些指标，要想获得这些指标，我们必须采用另外一种方式来完成，这里BASE就是解决这个问题的办法。

## 1.6分布式+集群简介

分布式系统（distributed system）

由多台计算机和通信的软件组件通过计算机网络连接（本地网络或广域网）组成。分布式系统是建立在网络之上的软件系统。正是因为软件的特性，所以分布式系统具有高度的内聚性和透明性。因此，网络和分布式系统之间的区别更多的在于高层软件（特别是操作系统），而不是硬件。分布式系统可以应用在在不同的平台上如：Pc、工作站、局域网和广域网上等。

简单来讲：

1、分布式：不同的多台服务器上面部署不同的服务模块（工程），他们之间通过Rpc/Rmi之间通信和调用，对外提供服务和组内协作。

2、集群：不同的多台服务器上面部署相同的服务模块，通过分布式调度软件进行统一的调度，对外提供服务和访问。