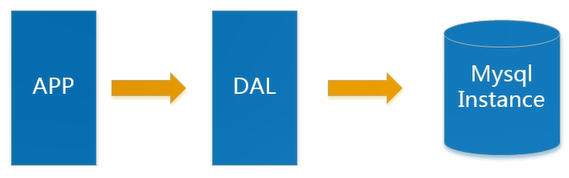
# 1.入门概述

## 1.1互联网时代背景下大机遇，为什么用nosql

### 1.1.1单机MySQL的美好年代

90年代，一个网站的访问量一般都不大，用单个数据库完全可以轻松应付。那个时候，更多的是静态网页，动态交互类型的网站不多。

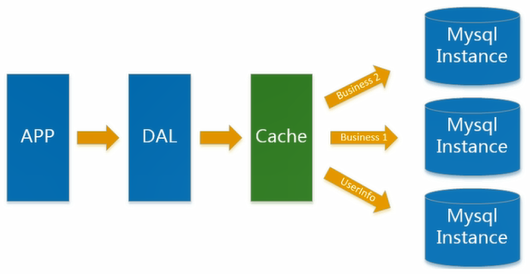


上述架构下，数据存储有以下一些瓶颈：

1. 数据量的总大小，一个机器放不下时；
2. 数据的索引（B+Tree），一个机器的内存放不下时；
3. 访问量（读写混合），一个实例不能承受。

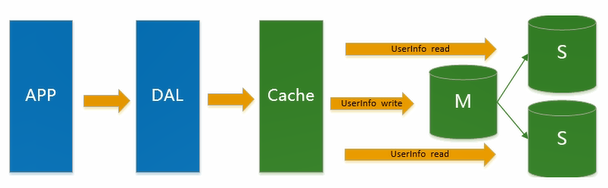
### 1.1.2 Memcached（缓存）+MySQL+垂直拆分

随着访问量的上升。几乎大部分使用MySQL架构的网站在数据库上都开始出现了性能问题，web程序不再仅仅专注在功能上，同时也在追求性能。程序员们开始大量使用缓存技术来缓解数据库的压力，优化数据库的结构和索引。开始比较流行的是通过文件缓存也带了比较高的IO压力。在这个时候，Memcached就自然的成为一个非常时尚的技术产品。



### 1.1.3 Mysql主从读写分离

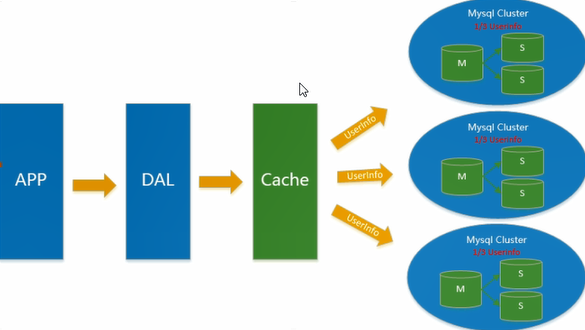
由于数据库的写入压力增大，Memcached只能缓解数据库的读取压力。读写集中在一个数据库上让数据库不堪重负，大部分网站开始使用主从复制技术来达到读写分离，以提高读写性能和读库的可扩展性。Mysql的master-slave模式成为这个时候的网站标配了。



### 1.1.4 分表分库+水平拆分+mysql集群

在Memcached的高速缓存，MySQL的主从复制，读写分离的基础之上，这时MySQL主库的写压力开始出现瓶颈，而数据量的持续猛增，由于MyISAM使用表锁，在高并发下会出现严重的锁问题，大量的高并发MySQL应用开始使用InnoDB引擎代替MyISAM。

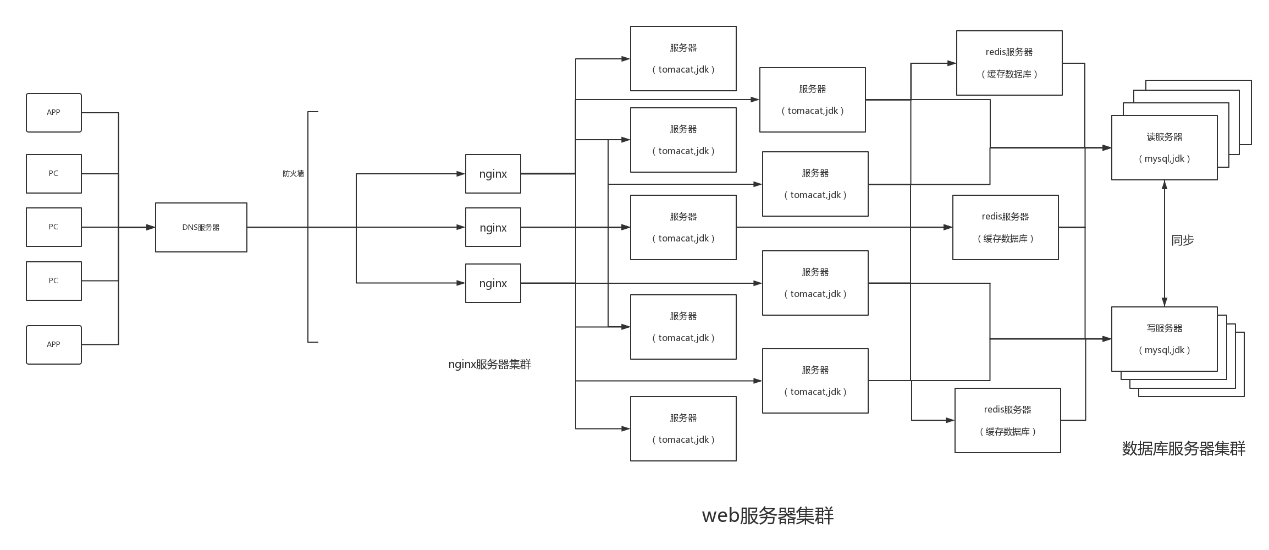
同时，开始流行使用分表分库来缓解写压力和数据增长的扩展问题。也就在这个时候，MySQL推出了还不太稳定的表分区。虽然MySQL推出了MySQL Cluster集群，但性能也不能很好满足互联网的要求，只是在高可靠性上提供了非常大的保证。



### 1.1.5 MySQL的扩展性瓶颈

MySQL数据库也经常存储一些大文本字段，导致数据库表非常的大，在做数据库恢复的时候就导致非常的慢，不容易快速恢复数据库。比如1000万4KB大小的文本就接近40GB的大小，如果能把这些数据从MySQL省去，MySQL将会变得非常的小。关系数据库很强大，但是它并不能很好的应付所有的应用场景。MySQL的扩展性差（需要复杂的技术来实现），大数据下IO压力大，表结构更改困难，正是当前使用MySQL的开发人员面临的问题。

### 1.1.6 今天是什么样子？



### 1.1.7 为什么用NoSQL

今天我们可以通过第三方平台（如Google，Facebook等）可以很容易的访问和抓取数据。用户的个人信息，社交网络，地理位置，用户生成的数据和用户操作日志已经成倍的增加。我们如果要对这些用户数据进行挖掘，那SQL数据库已经不适合这些应用了，NoSQL数据库的发展却能很好的处理这些大的数据。

