**传统关系数据库的瓶颈**  
传统的关系数据库具有不错的性能，高稳定型，久经历史考验，而且使用简单，功能强大，同时也积累了大量的成功案例。在互联网领域，MySQL成为了绝对靠前的王者，毫不夸张的说，MySQL为互联网的发展做出了卓越的贡献。  
在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。在那个时候，更多的都是静态网页，动态交互类型的网站不多。到了最近10年，网站开始快速发展。火爆的论坛、博客、sns、微博逐渐引领web领域的潮流。在初期，论坛的流量其实也不大，如果你接触网络比较早，你可能还记得那个时候还有文本型存储的论坛程序，可以想象一般的论坛的流量有多大。

**Memcached+MySQL**

  后来，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。程序员们开始大量的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来缓解数据库压力，但是当访问量继续增大的时候，多台web机器通过文件缓存不能共享，大量的小文件缓存也带了了比较高的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。

  Memcached作为一个独立的分布式的缓存服务器，为多个web服务器提供了一个共享的高性能缓存服务，在Memcached服务器上，又发展了根据hash算法来进行多台Memcached缓存服务的扩展，然后又出现了一致性hash来解决增加或减少缓存服务器导致重新hash带来的大量缓存失效的弊端。当时，如果你去面试，你说你有Memcached经验，肯定会加分的。

**Mysql主从读写分离**  
   由于数据库的写入压力增加，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。

**分表分库**

  随着web2.0的继续高速发展，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。这个时候，分表分库成了一个热门技术，是面试的热门问题也是业界讨论的热门技术问题。也就在这个时候，MySQL推出了还不太稳定的表分区，这也给技术实力一般的公司带来了希望。虽然MySQL推出了MySQL Cluster集群，但是由于在互联网几乎没有成功案例，性能也不能满足互联网的要求，只是在高可靠性上提供了非常大的保证。

**MySQL的扩展性瓶颈**

在互联网，大部分的MySQL都应该是IO密集型的，事实上，如果你的MySQL是个CPU密集型的话，那么很可能你的MySQL设计得有性能问题，需要优化了。大数据量高并发环境下的MySQL应用开发越来越复杂，也越来越具有技术挑战性。分表分库的规则把握都是需要经验的。虽然有像淘宝这样技术实力强大的公司开发了透明的中间件层来屏蔽开发者的复杂性，但是避免不了整个架构的复杂性。分库分表的子库到一定阶段又面临扩展问题。还有就是需求的变更，可能又需要一种新的分库方式。  
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。

**NoSQL概念**  
随着web2.0的快速发展，非关系型、分布式数据存储得到了快速的发展，它们不保证关系数据的ACID特性。NoSQL概念在2009年被提了出来。NoSQL最常见的解释是“non-relational”，“Not Only SQL”也被很多人接受。（“NoSQL”一词最早于1998年被用于一个轻量级的关系数据库的名字。）

NoSQL被我们用得最多的当数key-value存储，当然还有其他的文档型的、列存储、图型数据库、xml数据库等。在NoSQL概念提出之前，这些数据库就被用于各种系统当中，但是却很少用于web互联网应用。比如cdb、qdbm、bdb数据库。