**NOSQL的优势**

**易扩展**

NoSQL数据库种类繁多，但是一个共同的特点都是去掉关系数据库的关系型特性。数据之间无关系，这样就非常容易扩展。也无形之间，在架构的层面上带来了可扩展的能力。

**大数据量，高性能**

    NoSQL数据库都具有非常高的读写性能，尤其在大数据量下，同样表现优秀。这得益于它的无关系性，数据库的结构简单。一般MySQL使用Query Cache，每次表的更新Cache就失效，是一种大粒度的Cache，在针对web2.0的交互频繁的应用，Cache性能不高。而NoSQL的Cache是记录级的，是一种细粒度的Cache，所以NoSQL在这个层面上来说就要性能高很多了。

**灵活的数据模型**

  NoSQL无需事先为要存储的数据建立字段，随时可以存储自定义的数据格式。而在关系数据库里，增删字段是一件非常麻烦的事情。如果是非常大数据量的表，增加字段简直就是一个噩梦。这点在大数据量的web2.0时代尤其明显。

**高可用**  
NoSQL在不太影响性能的情况，就可以方便的实现高可用的架构。比如Cassandra，HBase模型，通过复制模型也能实现高可用。

**总结**

NoSQL数据库的出现，弥补了关系数据（比如MySQL）在某些方面的不足，在某些方面能极大的节省开发成本和维护成本。  
MySQL和NoSQL都有各自的特点和使用的应用场景，两者的紧密结合将会给web2.0的数据库发展带来新的思路。让关系数据库关注在关系上，NoSQL关注在存储上。  
 

1) 数据库表schema经常变化  
比如在线商城，维护产品的属性经常要增加字段，这就意味着ORMapping层的代码和配置要改，如果该表的数据量过百万，新增字段会带来额外开销（重建索引等）。NoSQL应用在这种场景，可以极大提升DB的可伸缩性，开发人员可以将更多的精力放在业务层。  
   
2)数据库表字段是复杂数据类型  
对于复杂数据类型，比如SQL Sever提供了可扩展性的支持，像xml类型的字段。很多用过的同学应该知道，该字段不管是查询还是更改，效率非常一般。主要原因是是DB层对xml字段很难建高效索引，应用层又要做从字符流到dom的解析转换。NoSQL以json方式存储，提供了原生态的支持，在效率方便远远高于传统关系型数据库。  
   
3)高并发数据库请求  
此类应用常见于web2.0的网站，很多应用对于数据一致性要求很低，而关系型数据库的事务以及大表join反而成了"性能杀手"。在高并发情况下，sql与no-sql的性能对比由于环境和角度不同一直是存在争议的，并不是说在任何场景，no-sql总是会比sql快。有篇article和大家分享下，http://artur.ejsmont.org/blog/content/insert-performance-comparison-of-nosql-vs-sql-servers  
   
4)海量数据的分布式存储  
海量数据的存储如果选用大型商用数据，如Oracle，那么整个解决方案的成本是非常高的，要花很多钱在软硬件上。NoSQL分布式存储，可以部署在廉价的硬件上，是一个性价比非常高的解决方案。Mongo的auto-sharding已经运用到了生产环境。http://www.mongodb.org/display/DOCS/Sharding  
 

并不是说NoSQL可以解决一切问题，像ERP系统、BI系统，在大部分情况还是推荐使用传统关系型数据库。主要的原因是此类系统的业务模型复杂，使用NoSQL将导致系统的维护成本增加。

**NoSQL和关系数据库结合**

        其实NoSQL数据库仅仅是关系数据库在某些方面（性能，扩展）的一个弥补，单从功能上讲，NoSQL的几乎所有的功能，在关系数据库上都能够满足，所以选择NoSQL的原因并不在功能上。所以，我们一般会把NoSQL和关系数据库进行结合使用，各取所长，需要使用关系特性的时候我们使用关系数据库，需要使用NoSQL特性的时候我们使用NoSQL数据库，各得其所。  
举个简单的例子吧，比如用户评论的存储，评论大概有主键id、评论的对象aid、评论内容content、用户uid等字段。我们能确定的是评论内容content肯定不会在数据库中用where content=’’查询，评论内容也是一个大文本字段。那么我们可以把 主键id、评论对象aid、用户id存储在数据库，评论内容存储在NoSQL，这样数据库就节省了存储content占用的磁盘空间，从而节省大量IO，对content也更容易做Cache。  
//从MySQL中查询出评论主键id列表 commentIds=DB.query("SELECT id FROM comments where aid='评论对象id' LIMIT 0,20"); //根据主键id列表，从NoSQL取回评论实体数据 CommentsList=NoSQL.get(commentIds);  
NoSQL代替MySQL

        在某些应用场合，比如一些配置的关系键值映射存储、用户名和密码的存储、Session会话存储等等，用NoSQL完全可以替代MySQL存储。不但具有更高的性能，而且开发也更加方便。

**NoSQL作为缓存服务器**  
       MySQL+Memcached的架构中，我们处处都要精心设计我们的缓存，包括过期时间的设计、缓存的实时性设计、缓存内存大小评估、缓存命中率等等。NoSQL数据库一般都具有非常高的性能，在大多数场景下面，你不必再考虑在代码层为NoSQL构建一层Memcached缓存。NoSQL数据本身在Cache上已经做了相当多的优化工作。  
    Memcached这类内存缓存服务器缓存的数据大小受限于内存大小，如果用NoSQL来代替Memcached来缓存数据库的话，就可以不再受限于内存大小。虽然可能有少量的磁盘IO读写，可能比Memcached慢一点，但是完全可以用来缓存数据库的查询操作。